AD-A253 492

TATION PAGE

Form Approved
OPM No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10 July 1992	Draft Handbook

(2)

4. TITLE AND SUBTITLE
Technical Report Work Breakdown Structure for Software Elements

5. FUNDING NUMBERS
N/A

6. AUTHOR(S)
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Ada Joint Program Office
The Pentagon, Rm 3E118
Washington, DC 20301-3080

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Ada Joint Program Office
The Pentagon, Rm. 3E118
Washington, D.C. 20301-3080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
MIL-HDBK-171

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; distribution unlimited

12b. DISTRIBUTION CODE
UNCLASSIFIED

13. ABSTRACT (Maximum 200 words)

MIL-HDBK-171 provides a framework for identifying and extending Work Breakdown Structure elements related to software when implementing DoD acquisition policy and MIL-STD-881B. This framework provides guidance to Industry and Government to better monitor, track, analyze, and estimate the cost of developing and supporting defense system software. The handbook shows how an acquisition can be planned and how contractor/developer software management information can be tailored to the specific requirements of the acquisition.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Ada Software, Costing, MIL-STD-881B	66
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

Technical Report Describing Contents of
MIL-HDBK-171 (DRAFT)
10 July 1992

TECHNICAL REPORT

WORK BREAKDOWN STRUCTURE ELEMENTS FOR SOFTWARE

FOREWORD

1. This military handbook is approved for use by all departments and agencies of the Department of Defense (DoD).

2. Beneficial comments (recommendations, additions, deletions) and any pertinent data that may be of use in improving this document should be addressed to Commander, U. S. Army Communications-Electronics Command (CECOM), ATTN: AMSEL-RD-SE-R-CRM, Fort Monmouth, NJ 07703-5000.

3. As Department of Defense (DoD) systems increase their reliance on computers, software complexity increases and its criticality grows. Computer software is a necessary element in DoD systems. The results of *Desert Storm* demonstrated the force multiplier that software provides. Modern management practices use work breakdown structures to help provide visibility into and manage risk involved with software developments. An organization that instruments its process, infrastructure, and quality will have better management control over resources and associated risks. The handbook shows how to achieve that objective through existing contractor cost/schedule control systems by properly structuring the contract WBS, statement of work WBS, and reporting requirements. The goals are to control software cost by improving software management and measurement techniques and to provide reliable performance reports to the Government at practical summary levels.

4. The Ada Joint Program Office determined that use of WBS for identifying and collecting software costs would be an effective management tool. As part of a U. S. Army Materiel Command (AMC) study on the defense software life cycle acquisition process, the *U. S. Army Materiel Command (AMC) Software Task Force Report, 28 February 1989,* recommended that an effective cost reporting and tracking process be established for controlling and managing software development and support costs. This study initiated the development of a WBS for software elements. Additionally, DoD emphasis was placed on more visible software costs including modification of MIL-STD-881, *Work Breakdown Structures for Defense Materiel Items,* reflecting software visibility requirements.

5. In March 1990, the Software Executive Officials (SEC) established a Software WBS Working Group. The charter, to review and develop a DoD handbook designed to formalize Software WBSs and recommend changes to MIL-STD-881. This handbook, titled *Work Breakdown Structure Elements for Software*, improves the situation by providing tailorable guidance through the use of the detailed software elements and the WBS common framework procedures outlined in MIL-STD-881B. This handbook assists the practitioner with the implementation of Public Law 102-172, Sec 8044, that provided, *effective July 1, 1992, all new Department of Defense procurements shall separately identify software costs in the work breakdown structure defined by MIL-STD-881 in those instances where software is considered a major category of cost.*

6. This handbook was developed by the Software WBS Working Group in coordination with the Joint Logistic Commanders Joint Policy Coordinating Group for Computer Resource Management and the AMC Software Task Force, and with the assistance of the U. S. Army CECOM Software Engineering Directorate and the Air Force Cost Analysis Agency. Active participation by the National Security Agency, the Defense Information System Agency, the Services, the Office of the Secretary of Defense, government software development contractors, and industry through the National Security Industrial Association and the Electronics Industries Association brought this high priority initiative to a successful conclusion.

Technical Report Describing Contents of
MIL-HDBK-171 (Draft)

CONTENTS

Technical Report Describing Contents of
MIL-HDBK-171 (Draft)

CONTENTS

CONTENTS

Technical Report Describing Contents of
MIL-HDBK-171 (Draft)

CONTENTS

Technical Report Describing Contents of

MIL-HDBK-171 (Draft)

1. SCOPE

1.1 Purpose. MIL-HDBK-171 (hereafter referred to as the handbook) provides a framework for identifying and extending Work Breakdown Structure elements related to software when implementing DoD acquisition policy and MIL-STD-881B. This framework provides guidance to Industry and Government to better monitor, track, analyze, and estimate the cost of developing and supporting defense system software. The handbook shows how an acquisition can be planned and how contractor/developer software management information can be tailored to the specific requirements of the acquisition.

1.2 Application.

1.2.1 Use. This handbook used in conjunction with MIL-STD-881B provides guidance for further identifying software costs in the WBS in those instances where software is considered to be a major category of cost. This handbook is a guide for both contractors and DoD components (government activities) to further define elements below the software element requirement established in MIL-STD-881B (*Work Breakdown Structure for Defense Materiel Items*[1]). MIL-STD-881B is the primary source for developing WBS.

1.2.2 Focus. Although the need for improved tracking and estimating of software costs has been widely recognized by DoD system acquisition managers and system development contractors, cost estimation focused on a WBS where software is considered to be a major part of the acquisition has not been treated formally. An effective tool for managing and tracking system development and support activities and related costs is through the WBS. In order to manage software costs, a WBS with emphasis on software must be clearly defined and accounted for throughout the system life-cycle process. The WBS approach currently defined in MIL-STD-881B for acquisition of defense materiel items accommodates this need by providing for visibility of software elements in defense systems acquisitions. This approach is comprehensive yet flexible so that it can be tailored and augmented with the appropriate software WBS elements depending upon the nature of the system acquisition.

1.3 Benefits. The primary benefits gained by using the software WBS elements framework and guidance herein are the following:

a. More effective management practices to reduce software risk.

b. Integration into existing cost/schedule control systems.

c. Linkage to statistical process control and quality management techniques throughout the software life cycle.

---

1. Defense materiel items will be referred to as defense systems for the remainder of this handbook

d. Influencing Research and Development (R&D) directions based upon lessons learned in areas such as technology shortfalls, labor-intensive activities, and environmental constraints.

e. Accumulation of software cost data to aid the system acquisition and engineering planning process including cost estimation, scheduling, and staffing.

f. Calibration or construction of formal software cost estimation models.

g. Development of a cost estimation management framework throughout the software life cycle for current and future requirements.

h. Provision to increase awareness of the significance and complexity of software activities and related costs.

i. Ability to measure and troubleshoot performance throughout the life cycle including schedule, costs, distribution of effort, and contractor performance capability.

1.4 Organization of the handbook. The handbook is organized as shown in Figure 1-1. Sections 1, 2, and 3 provide the scope, applicable documents, and definitions for the handbook.

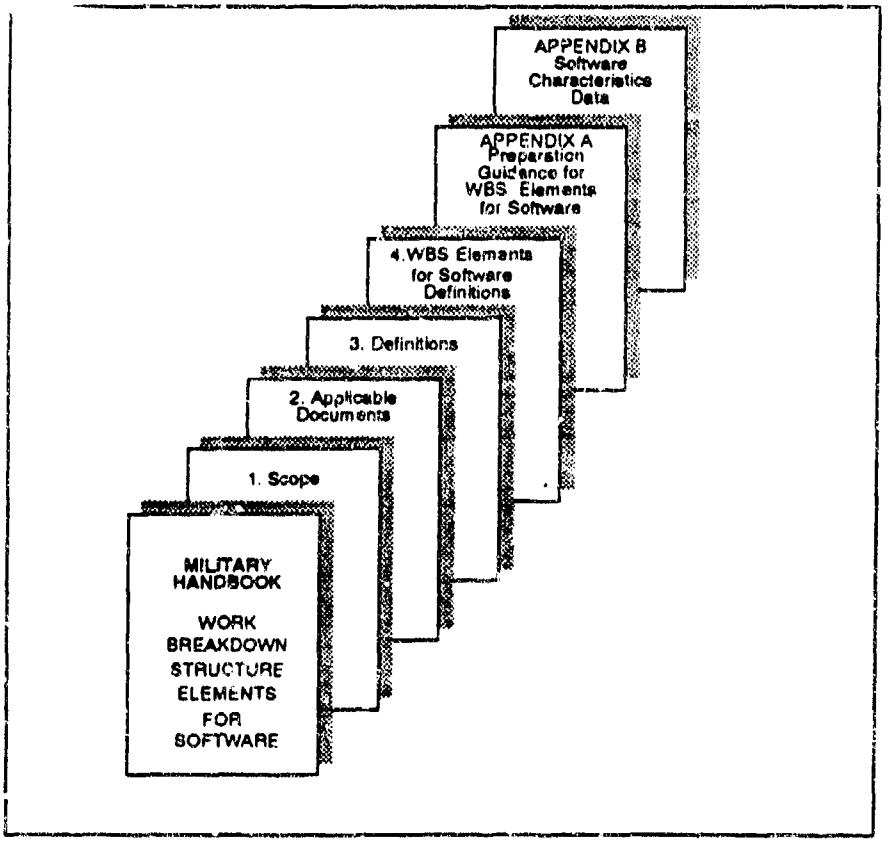


FIGURE 1-1. MIL-HDBK-171 Organization

1.4.1 **Section 4. Software WBS elements.** This section describes how a WBS accommodates software or software intensive programs. It describes in detail the WBS software elements and their relationship with each other, with subsystems, and with common WBS elements. Representative activities and deliverables for each element are provided as examples. This section also provides a brief discussion of how the WBS and associated performance measurement concepts can be used to enhance program and risk management.

1.4.2 **Appendix A. Preparation guidance for work breakdown structure elements for software.** This appendix identifies the importance of collecting software costs and provides examples of a WBS and the applicable approach to identify software elements as it relates to defense systems programs and contracts formulation. Sample WBS including software elements for a weapon system and an Automated Information System (AIS) are provided.

1.4.3 **Appendix B. Software characteristics data collection.** This appendix provides guidance for collecting data on project characteristics that collectively characterize the software being developed. These characteristics provide valuable data, which when collected and analyzed by the program manager (both Government and Industry), reinforces management indicators with credible software resource consumption estimates. A generalized questionnaire is provided that contains data items that are common to many existing software cost estimating models.

2. APPLICABLE DOCUMENTS

2.1 Government Documents.

2.1.1 Specifications, standards, and handbooks. The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the current issue of the Department of Defense Index of Specifications and Standards (DODISS).

STANDARDS

MILITARY

MIL-STD-881B	-	Work Breakdown Structures For Defense Materiel Items
DOD-STD-1467	-	Software Support Environment
DOD-STD-2167A	-	Defense System Software Development
DOD-STD-2168	-	Defense System Software Quality Program
DOD-STD-7935A	-	DOD Automated Information System (AIS) Documentation Standard

HANDBOOKS

MILITARY

MIL-HDBK-347	-	Mission-Critical Computer Resources Software Support
MIL-HDBK-782	-	Software Support Environment

(Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Standardization Documents Order Desk, 700 Robbins Avenue, Building #4, Section D, Philadelphia, PA 19111-5094.)

2.1.2 Other Government documents, drawings, and publications. The following other Government documents, drawings, and publications form a part of this document to the extent specified herein. Unless otherwise specified, the issues are those cited in the solicitation.

Technical Report Describing Contents of

MIL-HDBK-171 (Draft)

PUBLICATIONS

Contractor Cost Data Reporting (CCDR)[1]

AFLCP 800-15	-	Air Force Logistics Command Pamphlet
AFSCP 800-15	-	Air Force Systems Command Pamphlet
AMC-P 715-8	-	Army Materiel Command Pamphlet
NAVMAT P-5241	-	Navy Materiel Command Pamphlet

Cost/Schedule Control Systems Criteria Joint Implementation Guide[2]

AFCCP 173-5	-	Air Force Communications Command Pamphlet
AFLCP 173-5	-	Air Force Logistics Command Pamphlet
AFSCP 173-5	-	Air Force Systems Command Pamphlet
AMC-P 715-5	-	Army Materiel Command Pamphlet
NAVSO P3627	-	Assistant Secretary of the Navy (S&L) Pamphlet
DLAH 8400.2	-	Defense Logistics Agency Handbook
DCAA P7641.47	-	Defense Contract Audit Agency Pamphlet
DODI.5000.2	-	Defense Acquisition Management Policies and Procedures
DI-A-3023	-	Contract Work Breakdown Structure

(Unless otherwise indicated, copies of agency pamphlets are available from the Standardization Documents Order Desk, 700 Robbins Avenue, Building #4, Section D, Philadelphia, PA 19111-5094.)

2.2 **Non-Government Publications.** The following document(s) form a part of this document to the extent specified herein.

PUBLICATION

ANSI/IEEE Std 610.12-1990	Software Engineering Terminology
ANSI/IEEE Std 1016-1987	IEEE Recommended Practices for Software Design Description.

(Non - Government standards and other publications are normally available from organizations that prepare or distribute the documents. These documents also may be available in or through libraries or other informational services.)

1. This publication is one and the same document for all joint publication numbers listed.
2. Same as note 1.

3. ACRONYMS AND DEFINITIONS

3.1 Acronyms and definitions used in this handbook. The acronyms used in this handbook are defined as follows:

ACWP	-	Actual Cost of Work Performed
AIS	-	Automated Information System
ANSI	-	American National Standards Institute
BCWP	-	Budgeted Cost for Work Performed
BCWS	-	Budgeted Cost for Work Scheduled
CASE	-	Computer Aided Software Engineering
C/SCSC	-	Cost/Schedule Control Systems Criteria
C/SSR	-	Cost/Schedule Status Report
CCB	-	Configuration Control Board
CDR	-	Critical Design Review
CCDR	-	Contractor Cost Data Reporting
CDRL	-	Contract Data Requirements List
CI	-	Configuration Item
CM	-	Configuration Management
CFS	-	Contractor Furnished Software
CLIN	-	Contract Line Item Number
COTS	-	Commercial - off - the - shelf
CPR	-	Cost Performance Report
CSC	-	Computer Software Component
CSCI	-	Computer Software Configuration Item
CWBS	-	Contract Work Breakdown Structure
DID	-	Data Item Description
DoD	-	Department of Defense
DODISS	-	Department of Defense Index of Specifications and Standards
DSSE	-	Developmental Software Support Environment

ECP	-	Engineering Change Proposal
EIA	-	Electronics Industries Association
FCA	-	Functional Configuration Audit
FCHR	-	Functional Cost Hourly Report
FQR	-	Formal Qualification Review
FQT	-	Formal Qualification Test
GFE	-	Government Furnished Equipment
GFS	-	Government Furnished Software
HWCI	-	Hardware Configuration Item
ICD	-	Interface Control Document
IDD	-	Interface Design Document
IEEE	-	Institute of Electrical & Electronics Engineers
IRS	-	Interface Requirements Specification
KSLOC	-	Thousand Source Lines of Code
LL	-	Lower Levels
LCSSE	-	Life Cycle Software Support Environment
NOR	-	Notice of Revision
OT&E	-	Operational Test and Evaluation
PCA	-	Physical Configuration Audit
PDR	-	Preliminary Design Review
PDL	-	Program Design Language
PM	-	Project Manager
PMP	-	Prime Mission Product
POI	-	Program of Instruction
QDR	-	Quality Deficiency Report
R&D	-	Research and Development
RFP	-	Request for Proposal
SCN	-	Specification Change Notice
SDD	-	Software Design Document

SDF	-	Software Development File/Folder
SDL	-	Software Development Library
SDP	-	Software Development Plan
SDR	-	Systems Design Review
SDRB	-	Solicitation Data Review Board
SOW	-	Statement of Work
SPCR	-	Software Problem Change Report
SQL	-	Structure Query Language
SQPP	-	Software Quality Program Plan
SRR	-	System Requirements Review
SRS	-	Software Requirements Specification
SSA	-	Software Support Activity
SSPM	-	Software Standards and Procedure Manual
SSR	-	Software Specification Review
STR	-	Software Test Report
STD	-	Software Test Description
STP	-	Software Test Procedures
TQM	-	Total Quality Management
TRR	-	Test Readiness Review
VDD	-	Version Description Document
WBS	-	Work Breakdown Structure

3.2 Application software. ANSI/IEEE Std 610.12 - 1990: "Software designed to fulfill specific needs of a user; e.g., software for navigation, payroll, or process control. *Contrast with*: system software."

3.3 Common WBS elements. These are WBS elements that are applicable to all types of defense systems (e.g., data, training, system engineering/program management, peculiar support equipment, industrial facilities). The definition for the common WBS elements are provided in Section III of MIL-STD-881B.

3.4 Contract work breakdown structure (CWBS). MIL-STD-881B: "Contract work breakdown structure is defined as the complete work breakdown structure for a contract; i.e., the DoD approved work breakdown structure for reporting purposes and its discretionary extension to the lower levels by the contractor in accordance with this standard [MIL-STD-881B] and the contract work statement." This handbook generally is referring to CWBS.

3.5 Contractor. DODI.5000.2: "An entity in private industry that enters into contracts with the Government. In this instruction, the word may also apply to Government-owned, Government operated activities that perform work on major defense programs."

3.6 Cost analysis activity. The command activity responsible for cost estimating, monitoring, and providing assistance with cost reporting selection criteria. The cost analysis activity also provides assistance in developing contract peculiar clauses and data requirements selection.

3.7 Design entities. ANSI/IEEE Std 1016-1987. "A design entity is an element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced. Design entities result from a decomposition of the software system requirements. The objective is to divide the system into separate components that can be considered, implemented, changed, and tested with minimal effect on other entities". Design Entities include objects, CSC(s), modules, classes... etc..

3.8 Firmware. DOD-STD-2167A: "The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control."

3.9 Procuring activity. DODI.5000.2: "The subordinate command in which the Procurement Contracting Officer is located. It may include the program office, related functional support offices, and procurement offices. Examples are the Army Missile Command, Naval Sea Systems Command, and Air Force Electronic Systems Division."

3.10 Program work breakdown structure. MIL-STD-881B: "Covers the entire acquisition cycle of a program and consists of at least the first three levels of a WBS prescribed by MIL-STD-881B and its extension by the DOD component and/or contractor. The program WBS has uniform element terminology, definition, and placement in the family-tree structure. These levels have been organized within the following categories of defense materiel items:"

a. Aircraft systems
b. Electronic/Automated software systems
c. Missile systems
d. Ordnance
e. Ship systems.
f. Space systems.
g. Surface vehicle systems

3.11 Software support activity (SSA). MIL-HDBK-347: "The DoD or military service organization responsible for the software support of designated MCCRs" (Mission - Critical Computer Resources). This document expands on this definition to include non-mission critical computer resource systems which includes AIS.

3.12 System software. ANSI/IEEE Std 610.12 - 1990: "Software designed to facilitate the operation and maintenance of a computer system and its associated programs; e.g., operating systems, built in test, utilities which executes as part of the system. *Contrast with*: application software.'

3.13 Work breakdown structure (WBS). MIL-STD-881B: "A product-oriented family tree composed of hardware, *software*, services, data, and facilities which results from system engineering efforts during the development and production of a defense materiel item, and which completely defines the program. A work breakdown structure displays and defines the product(s) to be developed or produced and relates the elements of work to be accomplished to each other and to the end product."

4. WBS ELEMENTS FOR SOFTWARE DEFINITIONS

4.1 WBS overview. The work breakdown structure is the basis for communication throughout the acquisition process. It provides the common link unifying the planning, scheduling, cost estimating, budgeting, contracting, configuration management, and performance reporting disciplines. A typical work breakdown structure for a defense system composed of hardware, software, and common WBS elements (i.e., system engineering/program management and system test and evaluation, ... etc.) is illustrated by figure 4-1. The elements identified in this figure are represented generically. The actual element nomenclature would be specified for the particular defense system category; e.g., aircraft system, electronic system, or ship system.

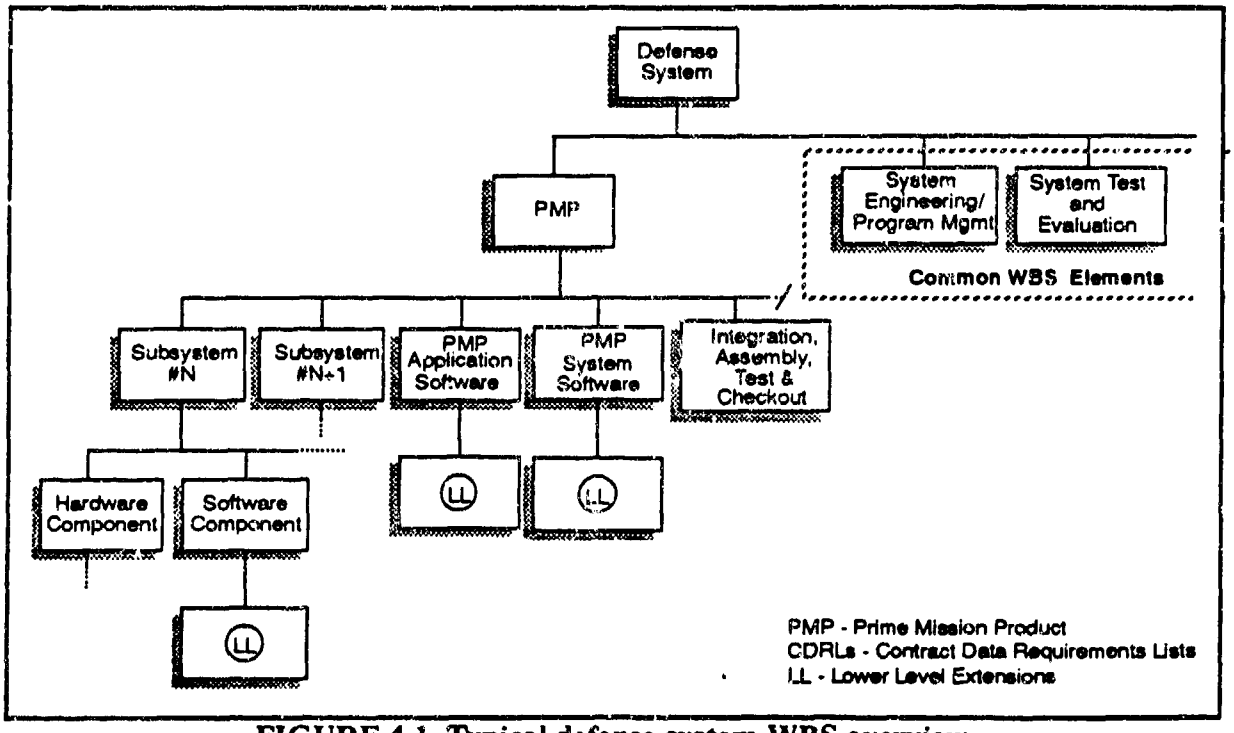


FIGURE 4-1. Typical defense system WBS overview

4.2 Defense System. The defense system encompasses all the Prime Mission Product (PMP) and common WBS elements for the specified defense system category as defined in MIL-STD-881B. The common WBS elements are those elements applicable to all types of defense systems. The common WBS elements, software elements and associated lower level extensions are further described in this handbook. In defense systems that consist of several subsystems, each subsystem must be identified separately with its associated hardware and software component. The actual name of each subsystem should be specified (i.e., "launch and guidance" or "navigation/guidance").

Subordinate to the PMP are the hardware subsystem and software (application and system) elements. Each subsystem (i.e., sensors, communications, or navigation/guidance) can be an aggregation of hardware and software component. Each WBS element for software is composed of one or more builds and is further defined as a Computer Software Configuration Item (CSCI). The software extension is further described in this handbook.

4.2.1 Prime Mission Product (PMP). The PMP element refers to the hardware and software used to accomplish the primary mission of the defense materiel item. It includes all integration, assembly, test and checkout, as well as all technical and management activities associated with individual hardware/software elements. Also included are integration, assembly, test and checkout associated with the overall PMP. When the electronic/automated software system comprises several PMPs, each will be listed separately at the same level. Also included are all whole and partial prime contractor, subcontractor, and vendor breadboards, brassboards, software prototypes and qualification test units. It also includes the design, development and production of complete units (i.e., the prototype or operationally configured units which satisfy the requirements of their applicable specification(s), regardless of end use). It excludes only those "less than whole" units (e.g., test, spares, etc.) consumed or planned to be consumed in support of system level tests. This element also includes factory special test equipment, special tooling, development software support environment, and production planning required to fabricate the PMP hardware and software. Duplicate or modified factory special test equipment and/or programming support environment delivered to the government for depot repair and post deployment software support is excluded and should be included in the peculiar support equipment element.

4.2.1.1 Subsystem. This element refers to all hardware and software components of the specific subsystem, including all associated special test equipment, special tooling, production planning, programming environments and all technical and management activities. The software component element consists of the applications and system software required to direct and maintain the specific subsystem. All effort directly associated with the remaining same level and the integration, assembly, test and checkout of these elements into the PMP is excluded.

4.2.1.1.1 Hardware Component. This element refers to the mechanical, electrical, and electronic items of the particular subsystem. The software component element refers to all software that is an integral part of the hardware subsystem specification or specifically designed and developed for subsystem test and evaluation.

4.2.1.1.2 Software Component. This element refers to all software that is an integral part of any specific subsystem specification, for example navigation/guidance, launch and control, radar, sensors, data displays, payroll and personnel. This excludes software that is specifically designed and developed for system test and evaluation. This software can be an aggregation of application and system software.

4.2.2 PMP software WBS elements. The following is a description of the PMP software WBS elements defined within the WBS framework. The software WBS elements are described generically and apply to each type of defense system specified in MIL-STD-881B. The associated activities and deliverables[1] for which the cost data are collected are listed with each software WBS description. All formal products and deliverables will be specified as per DD Form 1423, prepared in accordance with DOD-STD-2167A, DOD-STD-1467, or other applicable standards and tailored to the specific application. Although the lower level software elements are depicted as being associated with the PMP application software in figure 4-2, they are represented and described generically and apply as well to the respective lower level for the subsystem component software element and for the PMP system software element. A distinction is made between reporting and collecting data. This is disussed in section 4.3.1 and other sections. When reusable software such as commercial-off-the-shelf (COTS) or Government Furnished Software (GFS), or software specifically developed for reuse, are used throughout the system, the cost associated with the development, integration, and maintenance should appear only once and be noted in the appropriate software WBS element. (Note: COTS, GFS, and reused software costs associated with integration and test should be counted every time it is integrated and tested in the different applications. Costs should be counted only once for development and maintenance.)

4.2.2.1 PMP application software. Application software is defined as software that is specifically produced for the functional use of a computer system (Ref. ANSI/IEEE Std 610.12-1990). Examples are battle management, weapons control, and data base management. This element refers to all effort required to design, develop, integrate and checkout the PMP applications, builds and computer software configuration items (CSCIs), not including the non-software portion of PMP firmware development and production. This excludes all software that is an integral part of any specific hardware subsystem specification. It is important to note that all software that is an integral part of any specific hardware subsystem specification or specifically designed and developed for system test and evaluation should be identified with that system or subsystem. hen software is part of a system or subsystem, it may be appropriate to collect lower level information when it exists. In such cases, the structure defined in figure 4-2 should be used.

4.2.2.2 PMP system software. System software is defined as software designed for a specific computer system or family of computer systems to facilitate the development, operation, and maintenance of the computer system and associated programs (i.e., operating systems, compilers, and utilities (refer ANSI/IEEE Std. 610.12-1990). This element refers to all effort required to design, develop, integrate, and check out the system software including all software developed to support any PMP application software development. This excludes all software that is an integral part of a specific hardware subsystem specification or specifically designed and developed for system test and evaluation. The system software can consist of multiple software builds. The structure shown in figure 4-2 should be used when lower level information is desired.

---

1. Activities and deliverables listed are provided as typical examples only and do not preclude any contract dependent requirements (i.e., all DOD-STD-2167A data requirements).

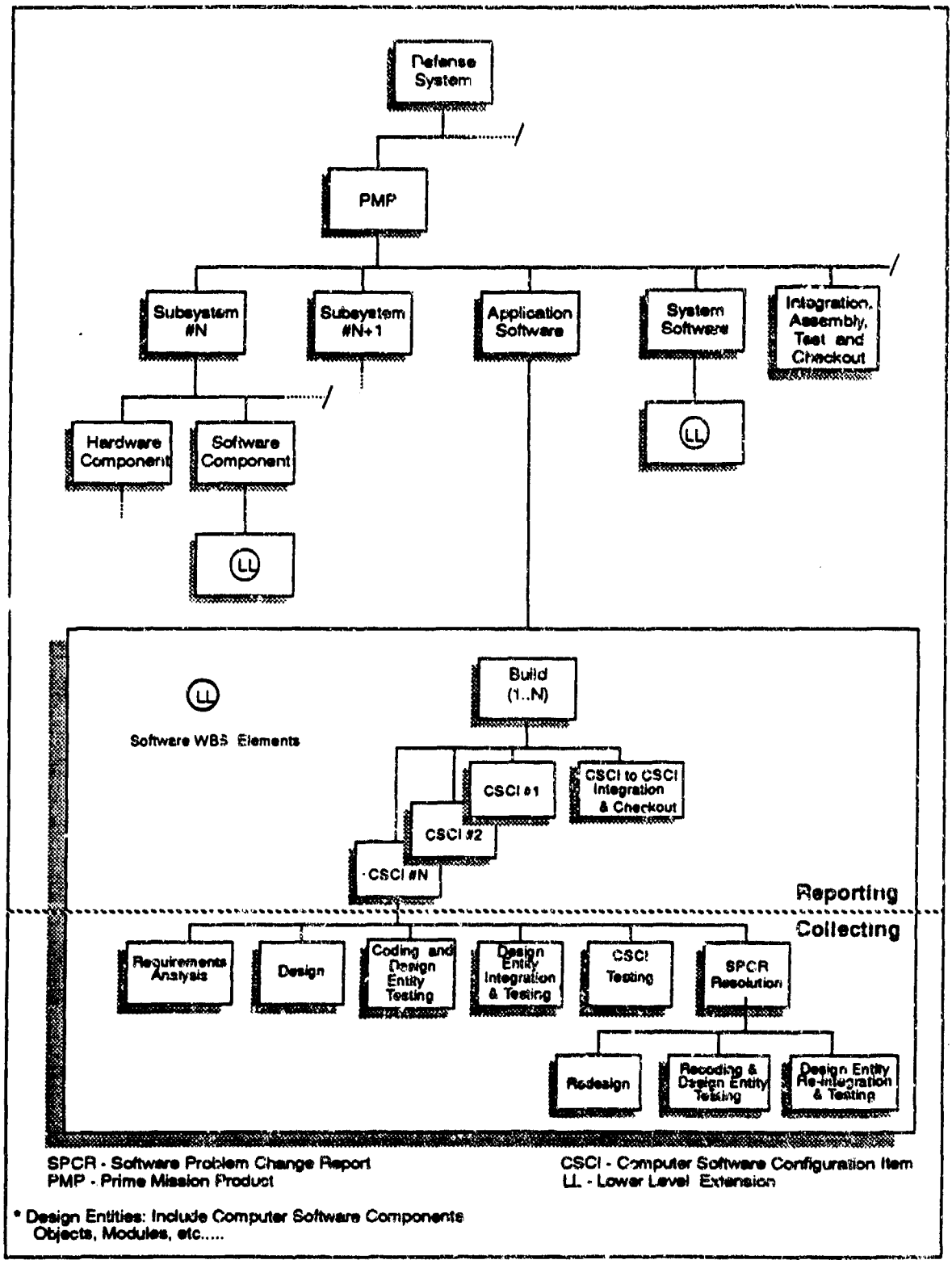


FIGURE 4-2. Lower level software WBS elements

4.2.3 Integration, Assembly, Test and Checkout. In those instances in which an integration, assembly, test and checkout element (figure 4-3) is used (Appendices A through G of MIL-STD-881B), it will include all effort of technical and functional activities associated with the design, development, and production of a development software support environment and support software required to assemble the equipment (hardware/software) elements into a prime mission product (hardware/software) as a whole and not directly part of any other individual element. Integration, assembly, test and checkout includes all effort associated with the following:

- Integration of PMP software
- The conduct of production (hardware/software) acceptance testing
- Acquisition and development of Development Software Support Environment (DSSE).

When an integration, assembly, test and checkout element is utilized at lower levels of the contract work breakdown structure, it will be summarized into the next higher level equipment (hardware/software) work breakdown structure element and should never be summarized directly into a level 3 integration, assembly, test and checkout element. In the case of software, the integration, test and checkout activity at the CSCI level and below should be included at the levels as shown in this handbook. The structure shown in figure 4-2 should be used when it is desired to collect lower level information.

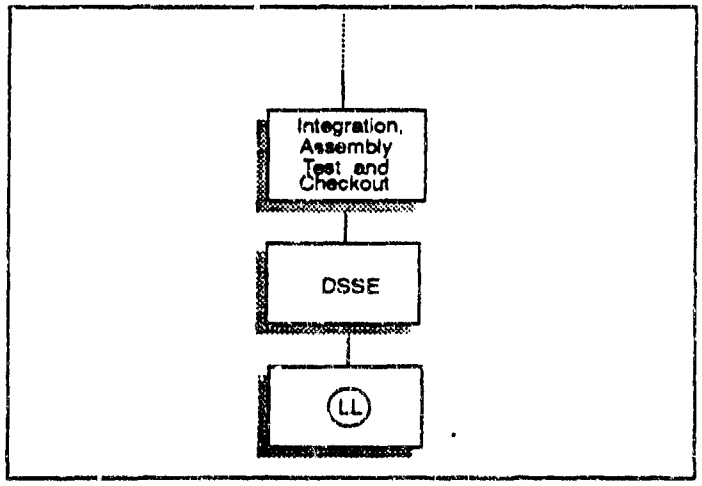


FIGURE 4-3. Integration, Assembly, Test and Checkout WBS

4.2.3.1 Development Software Support Environment (DSSE). This refers to all computer hardware and support software that is specifically developed or acquired to develop the mission software. Included in this element, e.g., are compilers, loaders, CASE tools, metric tools, software licenses and other computer resources that are determined to be essential in developing the mission software. The cost associated with this element is considered part of the CSCI it supports, or if more than one CSCI is involved it should be identified as a DSSE element and

included under the integration, assembly, test and checkout element. Included in this element is the cost for maintaining and operating the software reuse library.

If the DSSE equipment is to be transitioned to the Government for post deployment software support, the cost associated with the acquisition should be included in this element and only appear once. If this equipment is to be duplicated or modified, then the additional cost should be included in the Life Cycle Software Support Environment (LCSSE) element which is applied to the common element Peculiar Support Equipment.

4.3 Software development approach. Software development is a complex process, typically consisting of a number of parallel developments at the subsystem level. For many projects the development process is a series of parallel activities with several individual subsystem software products undergoing development at any one time. In a software development cycle lower level software elements (figure 4-2), can be replicated within a project many times depending on the complexity of development. The complexity of the development process can be managed and controlled by decomposing and organizing the system into smaller development subsystem software components or builds. The builds provide a snapshot of the system at any point in the acquisition process. This affords the developer an opportunity to gauge overall progress before the system is complete. The system can also be put into operation in increments or builds as a series of tested and delivered capabilities that are integrated throughout the system life cycle. These may be further organized into CSCIs, which are products that meet specific program requirements. The software WBS elements described in this handbook permit decomposition of the software project from a system to prime mission products.

In software development as with many engineering disciplines, the product decomposition below the CSCI level is performed by software professionals (such as designers, coders). The software product is a token that is passed along. Software is refined and elaborated from the software requirements analysis, the specification, the design, the code.

The initial partitioning of the requirements is the raw material from which the design and ultimately the code evolve. The requirements specification and design are integral to the software product. This is readily apparent when Ada program design language (PDL) is used. Software is the refinement and elaboration of the requirements and the design. *These lower levels do not imply a design methodology, but rather they are data taps into any process, (e.g., waterfall, spiral, incremental, evolutionary). These taps are methodology independent.* Figure 4-4 is an example of software WBS lower level extension related to a spiral model. The software WBS elements are primarily related to the software development process for a defense system. Examples of their associated Lower Level (LL) are described and delineated graphically in more detail throughout the following subparagraphs. This handbook also applies to the development or support of deliverable software to be implemented in firmware. The WBS software elements do not apply to the development of the hardware element of firmware.

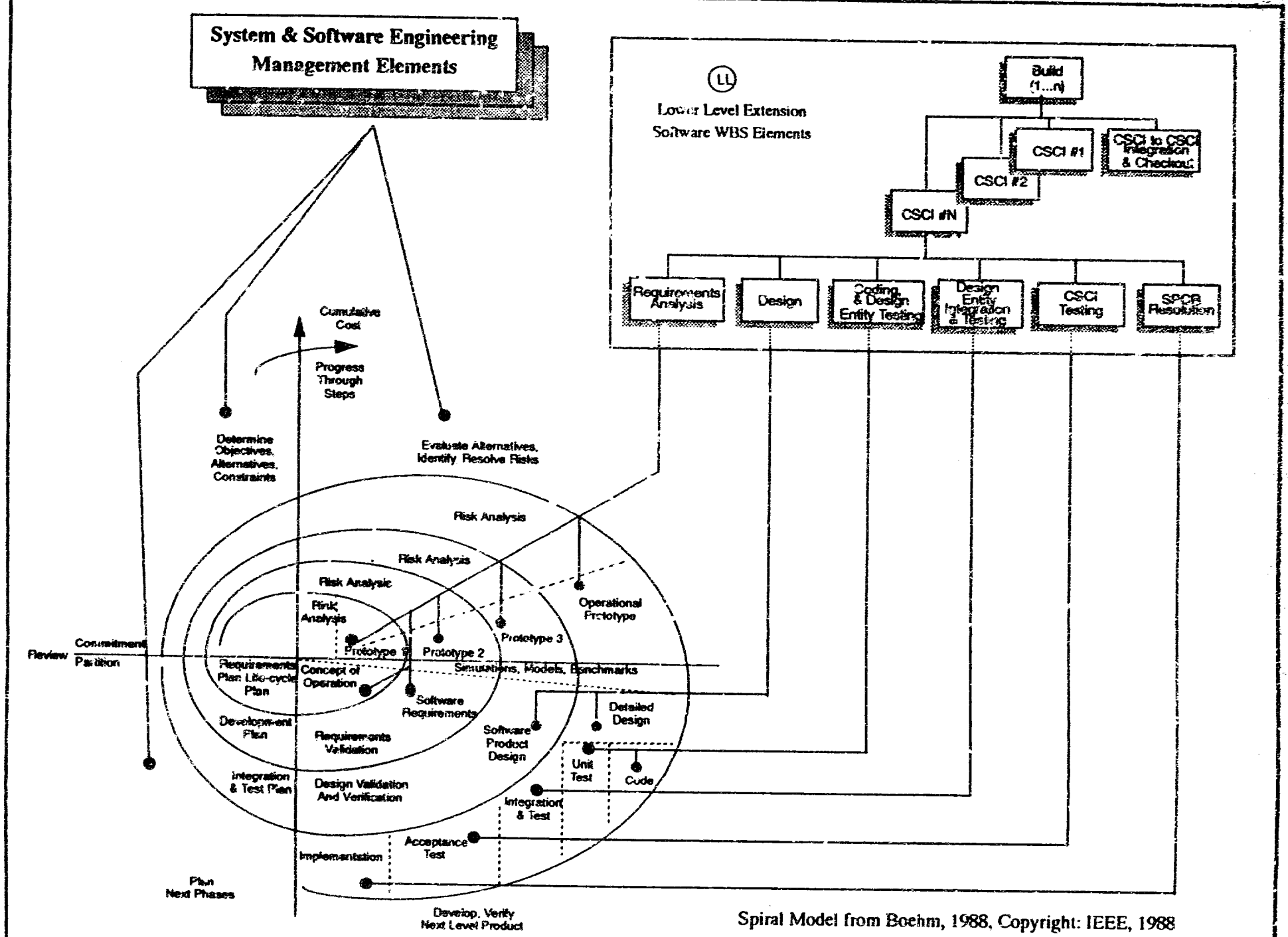


FIGURE 4-4. Notional taps into a spiral model

4.3.1 Software lower level extensions. The software element of the defense system can be further decomposed at the CSCI level and below as illustrated in figure 4-2. The software WBS elements, as identified in this handbook, provide insight into and consistency in software management information including cost and schedule tracking at or below the CSCI level. This handbook provides a common reference framework within which contractors developing or supporting software plan or report the cost data associated with elements of software development and support practices. It is recommended that contractors collect cost management information associated with software lower level elements. Software cost and schedule reporting will be to a level appropriate to the risk, cost, schedule, and interest of the program. The lower level extension for software WBS elements is used as a common collecting framework, consistent with DoD software development and acquisition policy.

4.3.2 Software build. A software build is an aggregate of one or more CSCIs that satisfies a specific set or subset of requirements based on development of software as defined in DOD-STD-2167A. When an incremental, spiral, or other software development method is used, multiple builds may be necessary to meet program requirements. A build is a separately tested and delivered product. Within builds are CSCIs. When a build is complete, a portion or all of one or more CSCI will be completed. Therefore, a CSCI may appear in more than one build, but will be successively more functional as each build is completed.

4.3.3 Computer software configuration item. An aggregation of software or any of its discrete portions that satisfy an end use function and have been designated by the Government for configuration management. CSCIs are the major software products of a system acquisition that are developed in accordance with DOD-STD-2167A. This includes reusable software components such as commercial off-the-shelf software, Government-furnished software, or software specifically developed for reuse. The CSCI element can be composed of design entities consisting of computer software components (CSCs) per DOD-STD-2167A, or objects, classes modules, etc. The distinct parts are functionally or logically identified by engineering decisions for convenience in designing and specifying a complex CSCI. It includes the effort associated with the requirements analysis, design, coding and testing, design entity integration and testing, CSCI testing, and software problem resolution of each CSCI.

4.3.3.1 Requirements analysis. This applies to the system requirements (as it pertains to software) and software requirement analysis activities as stated in DOD-STD-2167A. The software requirements analysis is the process by which a complete set of engineering and interface requirements are defined for each CSCI. Representative activities and deliverables are as follows:

Activities:

- Analysis of preliminary software requirements.
- Identification and allocation of software requirements into CSCIs.

- Analysis of preliminary interface requirements.
- Identification and resolution of interface requirements.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Formal minutes of review meetings, including presentation materials.
- Interface Requirements Specifications (IRS).
- Software Requirements Specification (SRS).
- Software requirements analysis technical reports.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.2 Design. Software design is the process of decomposing a high-level abstract requirement into lower level software elements. Preliminary design and detailed design activities are accomplished to map out high-level as well as low-level strategies for allocating requirements from Software Requirements Specifications (SRSs) and Interface Requirements Specifications (IRSs) for each CSCI to design entities, e.g. objects, classes, modules, CSC(s), etc. Representative sample activities and deliverables are as follows:

Activities:

- Creating and maintaining Software Development Files/Folders (SDFs)
- Analysis of preliminary software design.
- Derive and map out high (top) level software design specifications.
- Devise and map out low level (detailed) software design specifications.
- Analysis of preliminary interface design specification.
- Define and describe interface design specification.
- Generate input to software test planning.
- Formalize test requirements for design entities.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Formal minutes of review meetings, including presentation materials.
- Design analysis technical reports.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.3 Coding and design entity testing. Software coding is the process of creating a representation of the software design into a program language that may then be converted mechanically (i.e., by compilation) to an acceptable machine-executable representation. Each design entity is coded and subsequently tested to ensure that it satisfies its specific requirement. Representative activities and deliverables are as follows:

Activities:

- Maintaining SDFs.
- Coding and compiling activities.
- Conduct testing and analysis.
- Code walk-through activities.
- Performing compliance checks to coding conventions.
- Developing lower level design entities test and integration procedures.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Source code listings.
- SDFs.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.4 Design entity integration and testing. Integration or building of design entities into a CSCI is performed and testing of lower level threads is conducted to verify that algorithms and data employed in interfacing each design entity is correctly specified and implemented. This ensures that when combined into larger builds, they demonstrate compliance with stated customer/user requirements. Representative activities and deliverables are as follows:

Activities:

- Perform design entity integration analysis.
- Perform design entity build and lower level thread testing.
- Record test results.
- Perform dry run of formal qualification tests documented in Software Test Description (STD).

Deliverables: Formal deliverables are specified in DD Form 1423.

- Formal minutes of review meetings, including presentation materials.
- Integration and test analysis engineering notes.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.5 CSCI testing. CSCI testing is the process of demonstrating to the procuring activity that the CSCI can perform correctly under the full range of operating conditions specified in the requirements documentation and that each CSCI satisfies its specification requirements. Representative activities and deliverables are as follows:

Activities:

- Conduct formal qualification tests.
- Conduct test analysis and record test results.
- Generate input to the Software Test Reports (STRs).

Deliverables: Formal deliverables are specified in DD Form 1423.

- Formal minutes of review meetings, including presentation materials.
- Formal qualification test descriptions.
- Qualification tests analysis engineering notes.
- Software problem change reports.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.6 Software Problem Change Report (SPCR) resolution. SPCR resolution, also known as rework (figure 4-5), is the process of analyzing and correcting software problems. The problem may be as a result of logic deficiencies, non compliance with previously defined requirements and design specifications, or clarifications and other changes to previously defined requirements and design specifications. This reporting element begins once the software has been placed under development baseline control after the conduct of formal qualification testing but prior to Government/customer acceptance. The level of efforts related to redesign, recoding, and testing are included in the WBS elements described below. The level of efforts associated with the common WBS element activities such as software data, software quality assurance, and software configuration management incurred during the problem resolution process are included in the appropriate common WBS software elements.

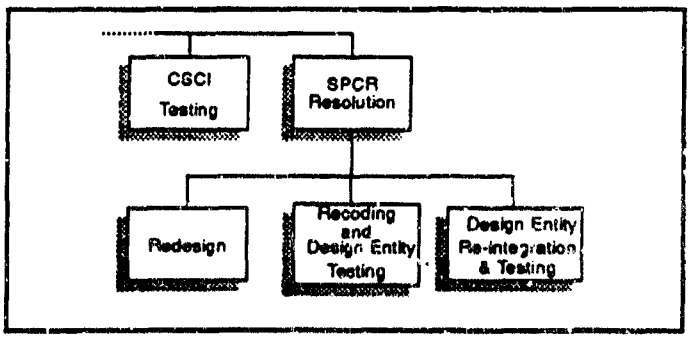


FIGURE 4-5. SPCR resolution WBS structure

4.3.3.6.1 Redesign. Redesign is the process of analyzing and modifying the software component abstractions based on the SPCR. This element includes the identification of deficiencies and reallocation of software component structures to comply with the previously defined requirements and specifications. Representative activities and deliverables are as follows:

Activities:

- Analyze the software problem report identified and reported.
- Make necessary corrections in design.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Updated design specifications.
- Design analysis engineering notes.
- Formal minutes of review meetings.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.6.2 Recoding and design entity testing. Recoding is the process of recreating the representation of the new or modified software design into the established programming language. All applicable design entities are recoded, tested and integrated to ensure that the identified problems are resolved and that the specific requirements are satisfied. Representative activities and deliverables are as follows:

Activities:

- Update software development files/folders (SDFs).
- Recoding and recompiling activities.
- Design entity testing, and analysis.
- Code walk-through activities.
- Performing compliance checks to coding conventions and software quality metrics.
- Update lower level entity test and integration procedures.

Deliverables: Formal deliverables are specified in DD Form 1423.

- New source code listings.
- Updated software development files/folders.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.3.6.3 Design entity reintegration and testing. Reintegration of applicable design entities is performed. Testing is once again conducted to verify that modifications to algorithms and data employed in interfacing each entity are correctly specified. Also, additional testing is conducted to ensure that modifications made to the software have not introduced new errors. Representative activities and deliverables are as follows:

Activities:

- Update input to software test planning.
- Redefine software test case as appropriate.
- Establish new test procedures where necessary.
- Perform design entity re-integration analysis.
- Maintain SDFs.

- Perform entity rebuild and retest.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Integration and test analysis engineering notes.
- Test documentation

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.3.4 CSCI to CSCI integration and checkout. This element includes integration and test verification and validation of the CSCIs. CSCI acceptance is performed and thread testing is conducted to verify that the algorithms and data employed for interfacing each CSCI are correctly specified and implemented so that they can demonstrate compliance with stated requirements. Representative activities and deliverables are as follows:

Activities:

- Generate input to software test plans, descriptions, and procedures.
- Define software test cases.
- Perform CSCI integration analysis.
- Perform software build and test.
- Update SDFs.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Formal minutes of review meetings, including presentation materials.
- Integration and test analysis engineering notes.
- Source code listing.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.4 Common work breakdown structure elements. These level 2 elements are common to all types of defense systems, as specified in MIL-STD-881B. The common elements encompass the areas of system engineering/program management, system test and evaluation, and training, etc. MIL-STD-881B groups several elements into a collection referred to as Common WBS (refer to figure 4-6). Examples of their associated level 3 and lower level software element extension, where applicable, are described and delineated graphically in more detail throughout the following subparagraphs.

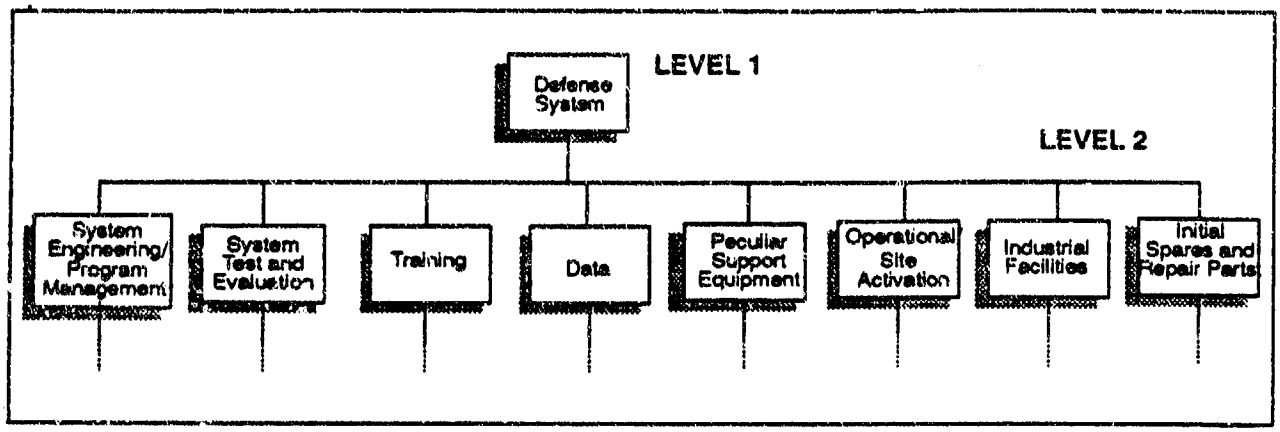


FIGURE 4-6. Common WBS elements

4.4.1 System engineering/program management. System engineering/program management (figure 4-7) refers to the software related activities associated with the overall system engineering, technical control, and business management of the particular program. This includes planning, directing, and controlling efforts of design engineering, reliability engineering, maintainability engineering, human factors engineering, logistics engineering, speciality engineering, software engineering, production engineering, and integrated test planning. The program management elements include planning, directing, coordinating, and monitoring the development and support efforts associated with the total system software.

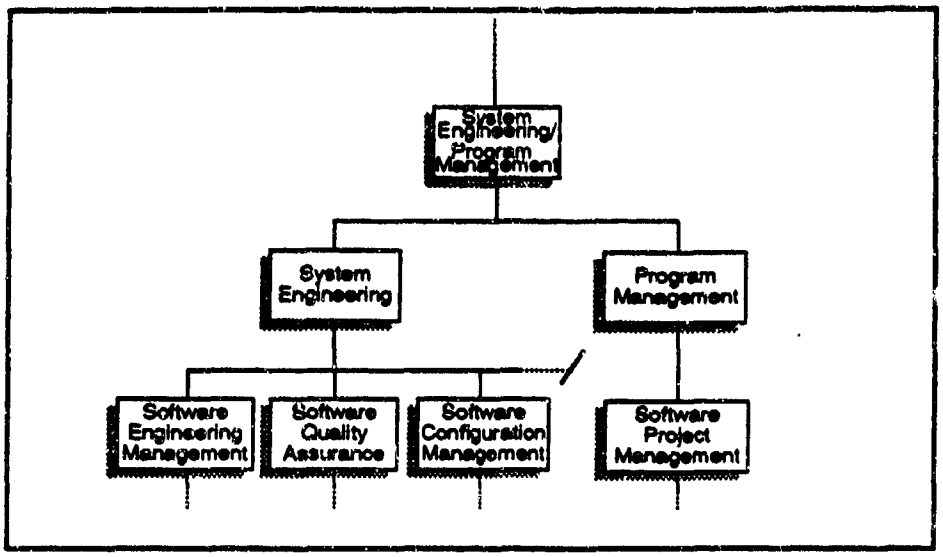


FIGURE 4-7. System engineering/program management WBS

4.4.1.1 Software project management. Software project management refers to the business and administrative planning, organizing, directing, coordinating, controlling, and approval actions designated to accomplish overall project objectives that are associated with specific software elements. Representative activities and deliverables are as follows:

Activities:

- Define and prepare overall task execution plans and project milestones.
- Monitor and direct various project activities.
- Prepare and report project progress and status.
- Control project cost-related activities.
- Prepare and report project funding expenditures.
- Direct and participate in all system and activity review meetings as necessary.
- Prepare and report software metrics data and status.
- Support & facilitate Government software review teams.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Prepare formal responses to Government queries.
- Prepare and submit progress reports.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.4.1.2 System engineering. The system engineering element is defined as the technical and management efforts of directing and controlling an integrated engineering effort of a system or

program. This element encompasses the system engineering effort to define the system and the integral planning and control of the technical program efforts of design engineering, specialty engineering, production engineering, and integrated test planning. This element includes, but is not limited to, the system engineering effort to transform an operational need or statement of deficiency into a description of system (hardware/software) requirements and a preferred system configuration and the technical planning and control effort for planning, monitoring, measuring, evaluating, directing, and replanning the management of the technical program. It specifically excludes the actual design engineering and the requirements analysis below the CSCI level. Representative activities and deliverables are as follows:

Activities:

- System level definition and requirements analysis.
- Design integrity analysis.
- Preparation of systems engineering management plan and specification.
- Program risk analysis.
- Participate and conduct technical reviews (e.g., system requirements reviews, system design reviews).

Deliverables: Formal deliverables are specified in DD Form 1423.

- Technical review of engineering notes.
- System engineering management plan.
- Technical reports and system engineering analysis notes.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.4.1.2.1 Software engineering management. The software engineering element refers to the technical and management efforts of directing and controlling the total software related engineering effort of a system. This includes planning and control of all defense system software engineering efforts, excluding the actual design, and some hardware engineering efforts to the extent of their direct relationship to and impact on the software engineering efforts. This element also includes analysis to determine the best allocation of system requirements between hardware, software, and personnel in order to partition the system into hardware configuration items (HWCIs) and CSCIs. Representative activities and deliverables are as follows:

Activities:

- Analyze system definition and requirements.
- Preparation and execution of Software Specification Reviews (SSRs).
- Prepare inputs to the systems engineering management plan.
- Define technical performance measurements.
- Participate in technical review meetings.
- Define intrasystem and intersystem compatibility.

- Perform risk analysis.
- Preparation and execution of software preliminary, critical, design reviews etc.
- Evaluate the impact of engineering change proposals (ECPs) on defense system software.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Formal minutes of review meetings, including presentation materials.
- Software engineering management plan.
- Technical reports and software engineering analysis notes.

For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.4.1.2.2 Software quality assurance. Software quality assurance includes those tasks that ensure compliance with the Government requirements for development and implementation of the contractor's software quality program. This program ensures the quality of (1) deliverable software and its documentation; (2) the processes used to produce deliverable software; and (3) non-deliverable software as specified in DOD-STD-2168. Representative activities and deliverables are as follows:

Activities:

- Generation or revision of the Software Quality Program Plan (SQPP)
- Participates in the Software Development Plan (SDP) development and review.
- Participation in Verification and Validation (V&V) support.
- Participation in formal reviews and audits.
- Conduct ongoing software quality evaluations of processes used in software development.
- Prepare and maintain records of software quality activities.
- Monitor the timeliness and appropriateness of software corrective actions.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Functional Configuration Audit (FCA) Report.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.4.1.2.3 Software configuration management. Software configuration management is a discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of CSCIs; (2) control changes to CSCIs and their related documentation; (3) audit CSCIs to verify conformance to specifications and Interface

Control Documents (ICDs); and (4) record and report change processing and implementation status information.

Activities:

- Conduct and maintain configuration status accounting.
- Develop baseline and version control for each CSCI.
- Participate in and provide input to FCA.
- Participate in and provide input to Physical Configuration Audit (PCA).
- Preparation of ECP.
- Preparation of Specification Change Notices (SCN).
- Generate inputs to the Software Configuration Management Plan.
- Preparation of Notice of Revisions (NORs)
- Generate inputs to Version Description Documents (VDDs).
- Participate in Configuration Control Board (CCB) meetings.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Configuration status accounting and engineering reports.
- Version control reports.
- Configuration management plan.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

4.4.2 **System test and evaluation.** The system test and evaluation element (figure 4-8) refers to the use of prototype, production, or specifically fabricated hardware/software to obtain or validate engineering data on the performance of the system during the development phase (normally funded from RDT&E) of the program. This element includes the detailed planning, conduct, support, data reduction and reports (excluding the Contract Data Requirements List (CDRL) data) from such testing, and all hardware/software items which are consumed or planned to be consumed in the conduct of such testing. It also includes all efforts associated with the design and production of models, specimens, fixtures, and instrumentation in support of the system level test program. NOTE: Test articles which are complete units (i.e., functionally configured as required by specifications) are excluded from this work breakdown structure element. **All formal and informal testing up through the subsystem level which can be associated with the hardware/software element are excluded. Acceptance testing is also excluded. These excluded efforts are to be included with the appropriate hardware of software elements.**

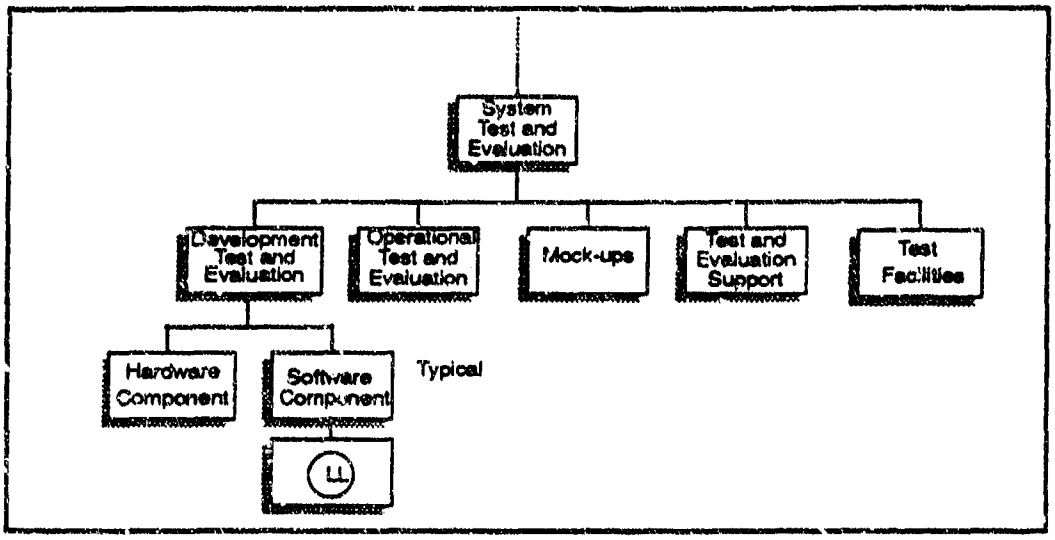


FIGURE 4-8. System test and evaluation WBS

4.4.2.1 **Development test and evaluation.** The development test and evaluation element refers to the test and evaluation conducted to: (1) demonstrate that the engineering design and development process is complete; (2) demonstrate that the design risks have been minimized; (3) demonstrate that the system will meet specifications; (4) estimate the system's military utility when introduced; (5) determine whether the engineering design is supportable (practical, maintainable, safe, etc.) for operational use; (6) provide test data with which to examine and evaluate trade-offs against specification requirements, life cycle cost, and schedule; and (7) perform the logistics testing efforts to evaluate the achievement of supportability goals, the adequacy of the support package for the system, e.g., deliverable maintenance tools, test equipment, technical publications, maintenance instructions, and personnel skills and training requirements etc. Development test and evaluation includes all contractor in-house effort and all effort of planning, conducting and monitoring by the developing agency of the DoD Component.

All programs, where applicable, include models, tests and associated simulations such as wind tunnel, static, drop, and fatigue; integration ground tests; test bed aircraft and associated support; qualification test and evaluation (QT&E), development flight test, test instrumentation, environmental tests, ballistics, radiological, range and accuracy demonstrations, test facility operations test equipment (including its support equipment), chase aircraft and support thereto, and logistics testing.

For aircraft, include avionics integration test composed of the following: (1) test bench/ laboratory, including design, acquisition, and installation of basic computers and test equipments which will provide an ability to simulate in the laboratory the operational environment of the

avionics system/subsystem; (2) air vehicle equipment, consisting of the avionics and/or other air vehicle subsystem modules which ar required by the bench/lab or flying test bed in order to provide a compatible airframe avionics system/subsystem for evaluation purposes; (3) flying test bed, including requirements analysis, design of modifications, lease or purchase of test bed aircraft, modification of aircraft, installation of avionics equipment and instrumentation, and checkout of an existing aircraft used essentially as a flying avionics laboratory; (4) avionics test program, consisting of the effort required to develop test plans/procedures, conduct tests, and analyze hardware and software test results to verify the avionics equipments' operational capability and compatibility as an integrated air vehicle subsystem; and (5) software, referring to the effort required to design, code, de-bug and document software programs necessary to direct the avionics integration test. The software in figure 4-8 is typical for each element. When appropriate the software component may be extended according to the structure shown in figure 4-2.

4.4.2.2 Operational test and evaluation. The operational test and evaluation element refers to that test and evaluation conducted by agencies other than the developing command to assess the prospective system's military utility, operational effectiveness, operational suitability, logistics supportability (including compatibility, inter-operability, reliability, maintainability, logistic requirements, etc.), cost of ownership, and need for any modifications. Initial operational test and evaluation conducted during the development of a weapon system will be included in this element. This element encompasses such tests as system demonstration, flight tests, sea trials, mobility demonstrations, on-orbit tests, spin demonstration, stability tests, qualification operational test and evaluation (QOT&E), etc., and support thereto, required to prove the operational capability of the deliverable system. It includes contractor support (e.g., technical assistance, maintenance, labor, material etc.) consumed during this phase of testing. It also includes performing the logistics testing efforts to evaluate the achievement of supportability goals; the adequacy of the support for the system. e.g., deliverable maintenance tools, test equipment, technical publications, maintenance instructions, and personnel skills and training requirements.

4.4.2.3 Mock-ups. The mock-ups element refers to the design engineering and production of system or subsystem mock-ups which have special contractual or engineering significance, or which are not required solely for the conduct of one of the above elements of testing.

4.4.2.4 Test and evaluation support. The test and evaluation support element refers to all support elements necessary to operate and maintain systems and subsystems during test and evaluation which are not consumed during the testing phase and are not allocated to a specific phase of testing. This element includes, i.e., repairable spares, repair of repairables, repair parts, warehousing and distribution of spares and repair parts, test and support equipment, test bed vehicles, drones, surveillance aircraft, tracking vessels, contractor technical support, etc., not

allocated to preceding test and evaluation elements. Operational and maintenance personnel, consumables, special fixtures, special instrumentation, etc., which are utilized and/or consumed in a single element of testing and which should, therefore, be included under that element of testing are excluded.

4.4.2.5 Test facilities. The test facilities element refers to those special test facilities required for performance of the various developmental tests necessary to prove the design and reliability of the system or subsystem. This element includes, i.e., test tank test fixtures, propulsion test fixtures, white rooms, test chambers, etc. The brick-and-mortar-type facilities identified as industrial facilities are excluded.

4.4.3 Training. The training element (figure 4-9) is defined as the deliverable training services, devices, accessories, aids, equipment, and parts used to facilitate instruction through which personnel will acquire sufficient concepts, skills, and aptitudes to operate and maintain the system with maximum efficiency. This element includes all effort associated with the design, development, and production of deliverable training equipment as well as the execution of training services. This element and its sub-elements specifically exclude the overall planning, management, and task analysis function inherent in the WBS element systems engineering/ management.

4.4.3.1 Software training services, equipment, and facilities. This element refers to the training services, special devices, equipment and facilities used to conduct instruction through which personnel will acquire sufficient concepts, skills, and aptitudes to operate and maintain the software portion of the system. This element includes the material, courses, and associated documentation development necessary to accomplish the contracted for objective of training, (primarily the computer software, courses, training aids, developed or constructed solely for the training mission). It encompasses the materials used for the purpose of acquainting the trainees with the system and equipment or establishing trainee proficiency. This element specifically excludes the deliverable training data associated with the WBS element, Support Data. Representative activities and deliverables are as follows:

Activities:

- Preparation and conduct of the following:
- Software installation training.
- Operational training involving host and target execution environment.
- Maintenance training involving the incorporation of software updates or changes.
- Training services provided for teaching software functional architecture.
- Development of unique training equipment.

Deliverables: Formal deliverables are specified in DD Form 1423.

- Training course and curriculum outlines.
- Instruction guides.
- Program of instruction (POI).
- Audio visual aids.
- Classroom training.
- Unique training devices.

(For the formal deliverables, costs associated with the administrative preparation, review, and distribution are excluded. These costs are included in the appropriate data WBS elements.)

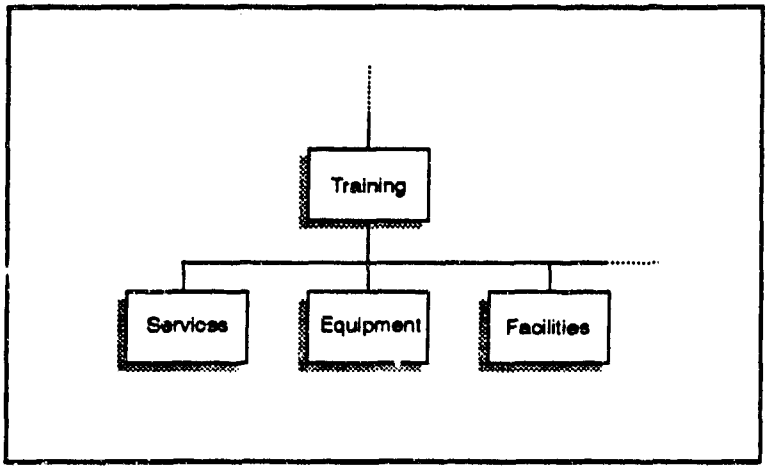


FIGURE 4-9. Training WBS

4.4.4 Data. The data element (figure 4-10) refers to all deliverable data required to be listed on a Contract Data Requirements List, DD Form 1423. The data requirements will be selected from the DoD Index of Specification and Standards (DODISS) and Acquisition Management Systems and Data Requirements Control List (AMSDL). This element includes only such effort that can be reduced or will not be incurred if the data item is eliminated. If the data are government peculiar, include the efforts for acquiring, writing, assembling, reproducing, packaging and shipping. Also included is the effort to transform, reproduce and ship data identical to that used by the contractor, but required in a different format by the Government.

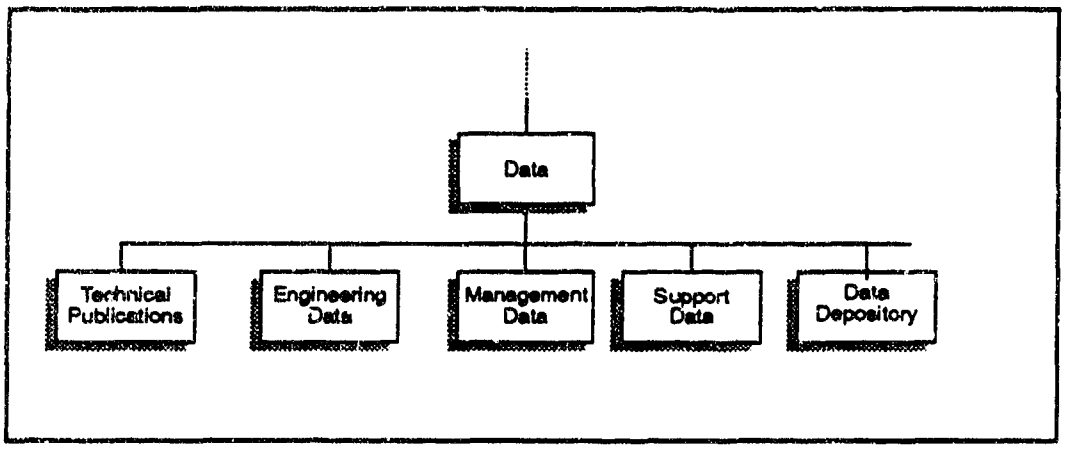


FIGURE 4-10. Data WBS

4.4.4.1 Software-related data. Software data refers to all required software-related documentation listed in DD Form 1423. It includes all data requirements in accordance with DOD-STD-2167A, and DOD-STD-1467, or other applicable standards tailored for the particular program. The cost data collected for this element are associated only with the preparation and review of documentation and do not include the efforts such as design, requirements analysis, and coding required in development of the software. Representative activities and deliverables are as follows:

Activities:

- Administrative efforts required to collect, review, prepare, and duplicate deliverable data in the format specified in appropriate Data Item Descriptions (DIDs).
- Administrative efforts required to complete and provide deliverable data to the Government in accordance with pertinent DIDs.

Deliverables: Formal deliverables are specified in DD Form 1423. Four examples of categories of data are shown below.

- Technical Publications: The technical publication element refers to those formal technical orders and manuals developed under the software development project.
- Engineering Data: The engineering data element refers to those engineering drawings, notes, and specifications produced during the project.
- Management Data: The management data element refers to those data items necessary for configuration management and cost, schedule, and project management undertaken during the project.
- Support Data: The support data element refers to those data items designed to document the logistics support planning.

4.4.4.2 Data depository. The data depository element is defined as a facility designated to act as custodian in establishing and maintaining a master engineering specification and drawing depository service for government approved documents that are the property of the U.S. Government. This element represents a distinct entity of its own and includes all effort of drafting, clerical, filing, etc.; required to provide the service. As custodian for the government, the contractor is authorized by approved change orders to maintain these master documents at the latest approved revision level. When documentation is called for on a given item of data retained in the depository, the charges (if charged as direct) will be to the appropriate data element. All similar effort for the contractor's internal specification and drawing control system, in support of its engineering and production activities, is excluded. The software data library is a repository for all software design records, documentation, source code, object code, and executable code associated with the defense system, produced and maintained during development (that is, the software development library) and throughout sustainment. Representative activities and deliverables are as follows:

Activities:

- Operation and maintenance of the software data library (including the software development library).
- Construction, modification, or expansion of the software data library

Deliverables: Formal deliverables are specified in DD Form 1423.

- Software data library content listing.
- Printed, microfilm, or magnetic media copies of documents, as required.

4.4.5 Peculiar support equipment. The peculiar support equipment element (figure 4-11) is defined to include the design, development, and production of those deliverable items and associated software required to support and maintain the system or portions of the system while not directly engaged in the performance of its mission, and which have application peculiar to a given defense materiel item. This element and its sub-elements **specifically exclude** the overall planning, management and task analysis functions inherent in the work breakdown structure element system engineering/program management, and common support equipment presently in the DoD inventory or commercially common within the industry which is bought by the using command and not by the acquiring command.

4.4.5.1 Support and handling equipment. The peculiar support and handling equipment element is defined as the deliverable tools and handling equipment used for support of the mission system, including the software. It typically includes ground support equipment, vehicular support equipment, powered support equipment, non-powered support equipment, munitions material handling equipment, materiel handling equipment. This element includes i.e., the Life Cycle Software Support Environment (LCSSE) required to maintain the mission software. Included as

an integral part of the LCSSE is any production or acquisition of duplicate or modified programming support environment, test and emulation equipment delivered to the procuring activity for use in maintaining the software.

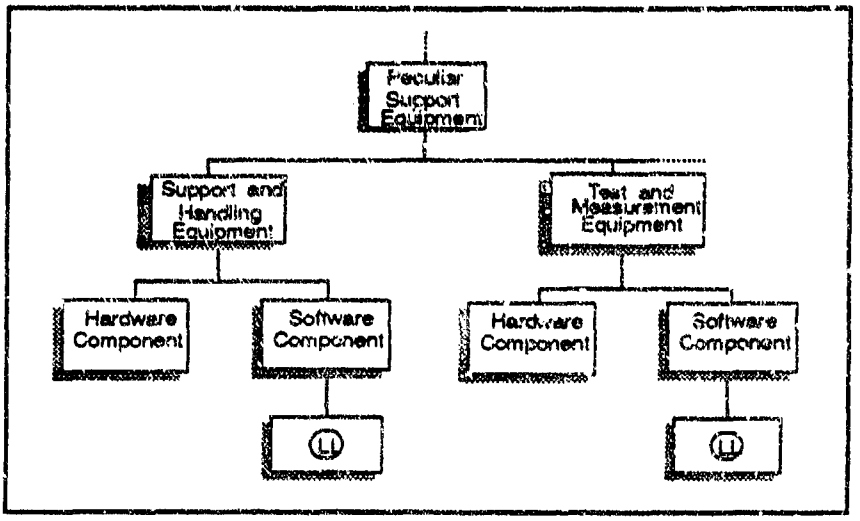


FIGURE 4-11. Peculiar support equipment WBS

Life cycle software support environment refers to all computer hardware and support software that is specifically developed or purchased commercially to aid the maintenance of the prime mission software. Included in this element, e.g., are compilers, loader, CASE tools, licenses, and other computer resources and utilities that are determined to be essential to support the software. If these items are acquired to develop the initial software and later the same items are to be provided to the procuring activity for post deployment software support, the associated cost will be included in the cost of producing the PMP and subsystem software. The structure shown in figure 4-2 should be used when it is desired to collect lower level information.

4.4.5.2 Test and measurement equipment. This element is defined as peculiar or unique testing and measurement equipment which allows an operator or maintenance function to evaluate operational conditions of a system or equipment by performing specific diagnostics, screening or quality assurance effort at an organizational, intermediate, or depot level of equipment support. It includes test measurement and diagnostic equipment, precision measuring equipment, automatic test equipment, manual test equipment, automatic test systems, test program sets, appropriate interconnect devices, automated load modules, tap(s) and related software, firmware and support hardware (power supply equipment, etc.) used at all levels of maintenance. It includes packages which enable line or shop replaceable unit, printed circuit

boards, or similar items to be diagnosed using automated test equipment. The structure shown in figure 4-2 should be used when it is desired to collect lower level information.

4.4.6 Operational/site activation. The operational/site activation element (figure 4-12) refers to the real estate, construction, conversion, utilities, and equipment to provide all facilities required to house, service, and launch prime mission equipment at the organizational and intermediate level. This element includes conversion of site, ship, or vehicle; system assembly, checkout, and installation (of mission and support equipment) into site facility or ship to achieve operational status. It also includes contractor support in relation to operational/site activation. This element includes efforts associated with operational/site activation of software maintenance and enhancements by the contractor based on software problem change reports and activities required for the release of new software build/versions as depicted if figure 4-2.

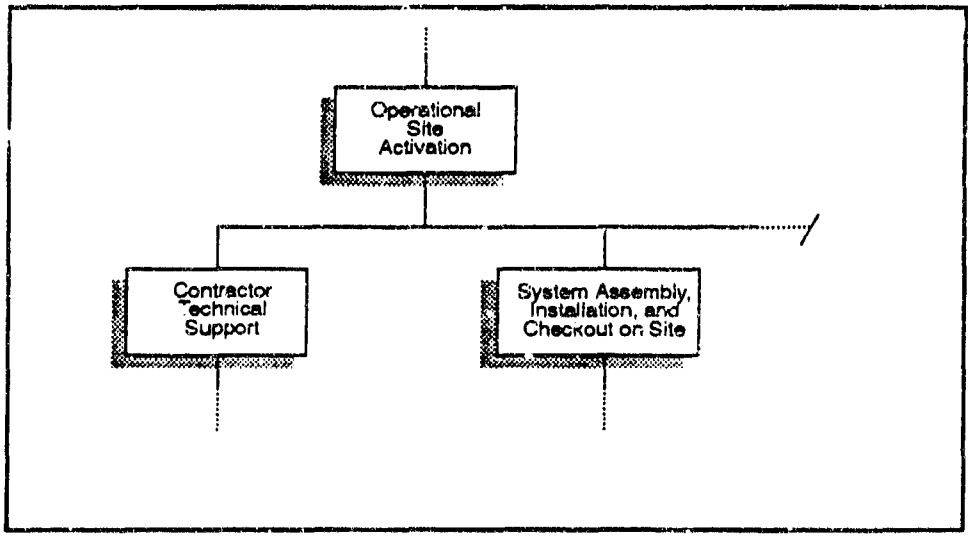


FIGURE 4-12. Operational /site activation WBS

4.4.6.1 Contractor technical support. The contractor technical support element refers to all materials and services provided by the contractor related to activation. This element includes repair of repairables, standby services, final turnover, etc. This also includes all efforts required to ensure that implemented and fielded software continues to fully support the operational mission of the software. This support begins once the baseline has been accepted by the procuring activity and continues until the responsibility for the support is transitioned to the Government software support activity. This element becomes critically important and its use more widely applied when software is developed under the concept of evolutionary, incremental or "build a little, test a little and field a little". Additional guidance relating to software support concepts, procedures,

and activities is detailed in MIL-HDBK-347. When software is being developed or maintained, it may be appropriate to collect lower level information when it exists. The structure shown in figure 4-2 and definitions should be used.

4.4.6.2 System assembly, installation, and checkout on site. The system, assembly, installation, and checkout on site element refers to the materials and services involved in the assembly of mission equipment at the site. This element includes; i.e., installation of mission and support equipment, operations, or support facilities; and complete system checkout or shakedown to insure achievement of operational status. Where appropriate, specify by site/ship or vehicle. As shown in figure 4-13, this element refers to the software product logistics support activities for the release of new software versions providing enhancements or corrections to deficiencies. Support activities also include the functions related to procurement, replication, distribution, and installation of technical materiel, to software media, and to personnel support required for the software version release.

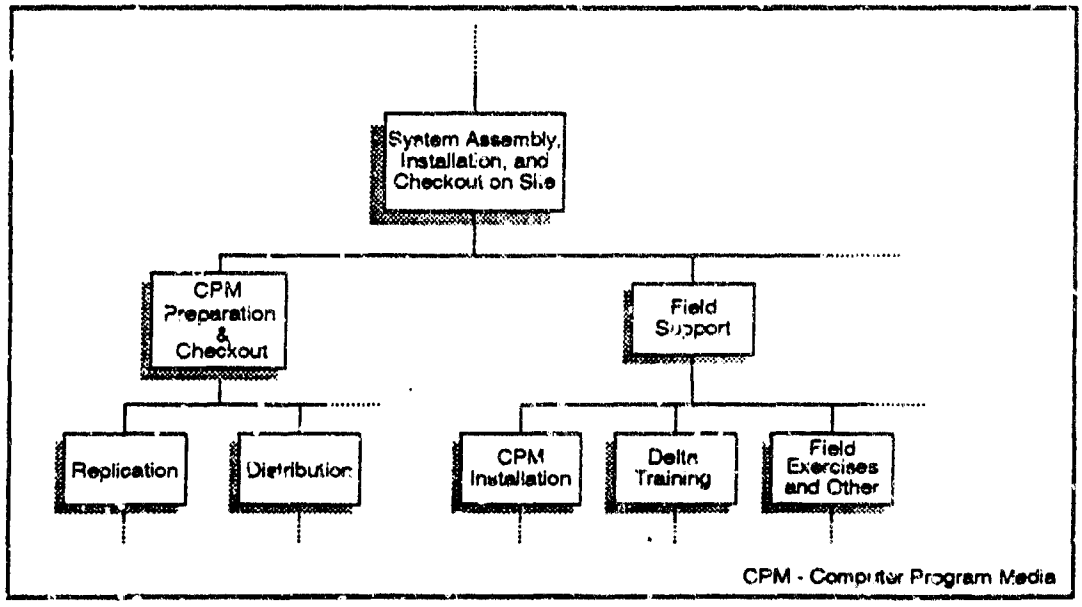


FIGURE 4-13. System assembly, installation, and checkout on site lower levels

4.4.6.2.1 Computer program media preparation & checkout. The computer program media element includes all activities required to produce the delivery packages, e.g., fabrication or initial production, replication or reprogramming, check-out, and distribution of the approved software version in all required deliverable forms of magnetic media, paper copy, and firmware memory devices that were duplicated or modified. This element excludes the initial development of the

CPM. The initial effort and costs are associated with the integration assembly, test and checkout of the PMP. Representative activities include the following:

- Production and check-out of all delivery media package items.
- Delivery coordination and distribution database maintenance.
- Coordination with hardware component modification efforts.
- Preparation of media installation instructions and version release user's guide.

4.4.6.2.2 **Field support.** Field support refers to the activities that assist field personnel (if necessary) in accomplishing successful program loading, updating, and operation of the new software version for the associated fielded defense system. This element includes activities associated with any problems and resolutions encountered, or training to the user community, as a result of the software change release package. Representative activities include the following:

- Computer program media installation.
- Providing support during field exercises.
- User instruction and instructor delta training for new functional capabilities, man-machine interfaces, and computer-aided instruction software.
- Identifying and resolving computer program media data release package discrepancies.
- Resupply of computer program media.

4.4.7 **Industrial facilities.** The industrial facilities element (figure 4-14) refers to the construction, conversion, or expansion of industrial facilities for production, inventory, and contractor depot maintenance (i.e., Post Deployment Software Support Activities) required by one or more suppliers for the specific system. This element includes; i.e., equipment acquisition, or modernization, where applicable, and maintenance of the above facilities or equipment. This element also includes all facility resource requirements associated with the software support activity (i.e., the expansion and modernization costs associated with maintaining the software facility and system test bed and with operating the LCSSE. This element excludes the acquisition cost of the LCSSE.

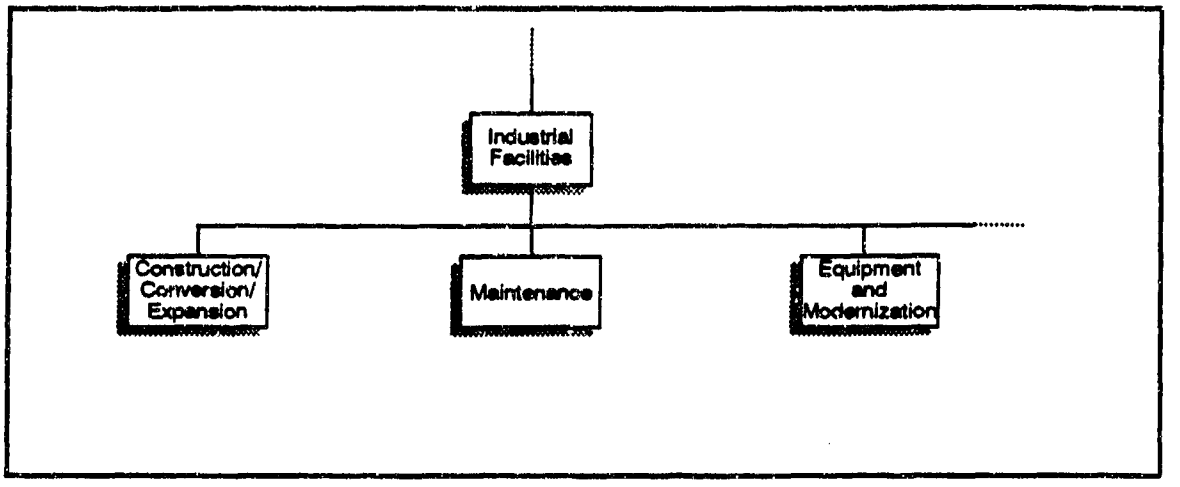


FIGURE 4-14. Industrial facilities WBS

4.4.8 **Initial spares and repair parts.** Initial spares and repair parts element is defined as the deliverable spare components, assembly and subassemblies used for initial replacement purposes in the materiel system equipment end item. This element includes the repairable spares and repair parts required as initial stockage to support and maintain newly fielded systems or subsystems during the initial phase of service, including pipeline quantities, at all levels of maintenance and support. This element excludes development test spares and spares provided specifically for use during installation, assembly and checkout on site.

4.5 **WBS relationship to program and risk management.** When extending the WBS with emphasis on software, the developer is not precluded from decomposing software beyond the CSCI element identified in MIL-STD-881B into design entities. The approach presented here for software is consistent with the DoD Cost/Schedule Control System Criteria (C/SCSC) requirements and with the Cost Performance Reports and the Cost/Schedule Status Report. The C/SCSC and related reports use the work breakdown structure as defined in the contract and extended by the contractor. A cost account manager is typically assigned at the point where the contractor organization and WBS intersect. The cost account manager generally builds work packages and typically has the responsibility of managing, monitoring and reporting variances. The resulting combination of well defined, short timespan work packages, estimate refinements, and status report monitoring provides vital management visibility into a program, increasing the chance of success. A correctly developed WBS and contractor performance measurement baseline with timely reviews reduces risk.

Risk management can be further enhanced by using the earned value performance management concept. Earned value is an objective measure of how much work has been accomplished. Without earned value, one can only compare how much has been spent with what was planned to be spent, with no objective indication of how much of the planned work was actually accomplished. Comparison between earned value (i.e., Budgeted Cost for Work Performed (BCWP)) and planned value (i.e., Budgeted Cost for Work Scheduled (BCWS)) provides an indication of whether or not schedule is being maintained. Comparison of earned value and actual costs (i.e., Actual Cost of Work Performed (ACWP)) provides information on whether costs are overrunning or underrunning. In turn, such comparisons allow trend analyses and evaluation of estimated costs at completion for all levels of the contract.

Earned value measurement requires a good Performance Measurement Baseline (PMB). A PMB is a time-phased budget plan against which performance is measured. The key to a good PMB is scheduling work in a way that is compatible with risk and meaningful in terms of the program milestones and technical requirements.

Since software is primarily a labor intensive, people oriented activity, cost and schedule variances usually relate directly to manpower loading. Manpower loading deviations from the plan are typically an early indicator of problems. Estimate updates utilizing the information collected and associated with identified characteristics (suggested guidance is given in Appendix B) provide a way of forecasting future performance. Thus, comparing the performance efficiency experienced to date with the remaining work budgets and estimates to complete provides an independent check of the reasonableness of the remaining Budgeted Cost for Work Scheduled (BCWS) and the projected final cost of the project. For a more detailed discussion of these concepts, see the *Cost Schedule Control Systems Criteria Joint Implementation Guide.*

4.5.1 **Contract reporting considerations.** Care must be exercised in establishing a reasonable WBS level for reporting to the government. Each lower level increases the number of reporting elements, with a commensurate increase in reporting burden and cost. Except for high-cost or high-risk elements and known problem areas, the level selected should usually be no lower than Contract Work Breakdown Structure (CWBS) CSCI level. Appropriate reporting elements below the CSCI level should be negotiated for each contract. While this handbook establishes a common framework for defining elements below the CSCI, it is not to be interpreted as a mandate to require routine reporting at or below the CSCI level.

APPENDIX A

PREPARATION GUIDANCE FOR

WORK BREAKDOWN STRUCTURE ELEMENTS FOR SOFTWARE

10. GENERAL.

10.1 Scope. The purpose of this appendix is to identify the importance of collecting software costs; discuss software costs reporting objectives and cost reporting; and provide examples of work breakdown structure for software as it relates to defense weapon system and Automated Information System (AIS) programs and contract formulation. The information contained herein is intended for guidance in compiling software cost data to satisfy applicable standards and acquisition policy. The procedures outlined in this Appendix are what is typically used by the DoD. MIL-STD-881B is the primary source for developing a WBS. The overall WBS acquisition strategy is developed relying heavily on MIL-STD-881B, specifically section II, user's guide.

10.2 Collecting software cost. It has been widely recognized that a majority of cost overruns encountered on systems with embedded computers are associated with the software. Software development time and cost most often exceed estimates by significant amounts. Lack of software-specific guidelines on cost tracking by the Government coupled with gross, lump-sum estimation of the software development cost as a whole by the contractor and bundling of software development costs into total system development costs has generated risk and management problems that a Work Breakdown Structure for Software will reduce. When software is considered to be a major category of cost, the contract should be written to ensure software is managed properly and that adequate software cost reporting is obtained. In view of software's increasing importance, contractors should be encouraged to adopt state-of-the-practices in software development, incorporating those practices into their management control and reporting systems.

The WBS approach permits the developer and maintainer to focus on the end products. Implementing proven methods of statistical process control at the measuring points identified by the WBS will permit management to identify insufficient resources, inefficient segments in the process, and other deficiencies both in the functional areas and in the development process. Implementing adequate management control systems cost and schedule and cost performance reporting, can enable an organization to influence its destiny and profits by managing its performance and productivity.

20. APPLICABLE DOCUMENTS. Guidance regarding cost/schedule control systems criteria information and instructions for preparing the various related reports are detailed in the references listed in section 2.1.2, **Other Government documents, drawings, and publications.**

30. PREPARATION INSTRUCTIONS

30.1 Pre Request for Proposal (RFP) Release.

30.1.1 Acquisition plan. During the preparation of the acquisition plan, the procuring activity normally will identify both the cost reporting items that will be required in each contract and the contract WBS elements to be reported against. Although cost reporting data requirements are established during the preparation of the procurement package, it is prudent practice to consider these requirements during acquisition planning. Figure A-1 provides a cost reporting chronology which lists activities required from acquisition planning through contract award. As part of this effort the procuring activity will prepare a Contractor Cost Data Reporting (CCDR) plan. This involves assignment of cost reporting needs and selection of the data elements that will be reported against in the particular reports. If further guidance or clarification is required, the Cost Analysis Activity can assist with cost report selection and the completion of the CCDR plan. The Software Support Activity is prepared to assist in developing the WBS including the software related elements.

30.1.2 Data call and solicitation data review board. During this next phase, the procuring activity must coordinate the data call with the Software Support Activity and the Cost Analysis Activity. The Software Support Activity can provide assistance in updating the WBS and the associated software elements, described in Section 4, which should be included in the Statement of Work (SOW). The Cost Analysis Activity has sample Contract Data Requirements Lists (CDRLs), DD Form 1423, and Statements of Work (SOWs) available. The Cost Analysis Activity can also provide assistance in developing contract peculiar clauses.

30.1.3 Identify software WBS elements. The next step is to identify those sub-systems in which software will be a critical or high risk item. Software is identified in two ways for the development of a work breakdown structure: (1) that which is part of the defense system/sub-system it supports, and (2) that which is stand alone (defense materiel items which have fully automated capabilities) and may support its own program or contract work breakdown structure. Figures A-2, A-3 and A-4 provide examples of how software should be addressed as part of the defense system/sub-system and as a stand alone pure software development/upgrade effort. The WBS for stand alone software system refers to the aggregation of software and appropriate common WBS elements required to develop and to produce a software capability for a command and control system, radar, stock point system and information system, etc.

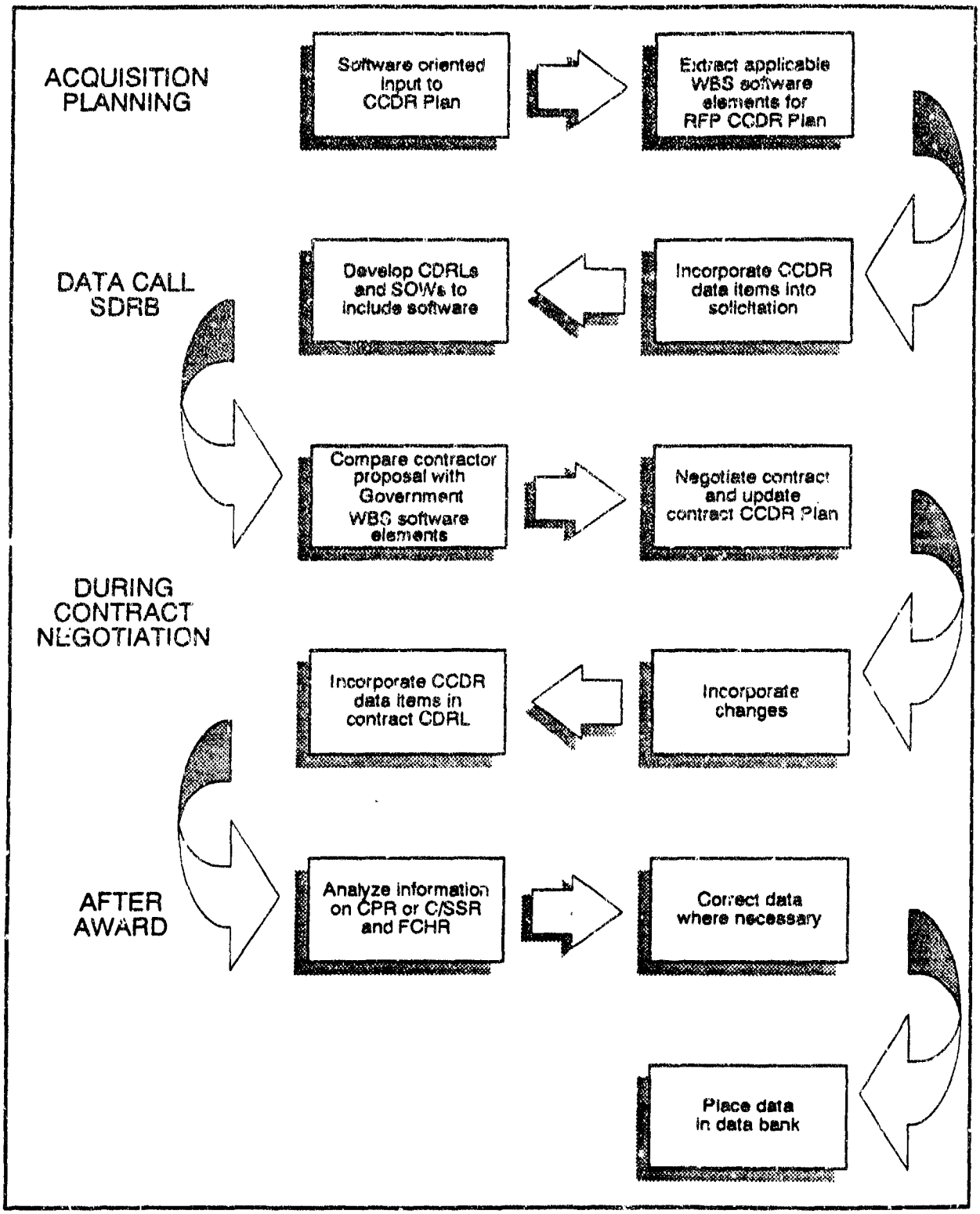


FIGURE A-1. Cost reporting chronology

30.1.4 Sample work breakdown structure.

30.1.4.1 Weapon system. For this example, the Navigation Guidance element is broken out as a separate item in accordance with appendix B (Electronics/Automated Software System) of MIL-STD-881B and augmented with software WBS elements discussed in this handbook, refer to Figure A-2.

30.1.4.2 Automated information system (AIS). For this example, a Command and Control AIS system is chosen, refer to Figure A-3. The Command and Control Automated Information System is broken out in accordance with Appendix B (Electronics/Automated Software System) of MIL-STD-881B and augmented with software WBS elements discussed in this handbook.

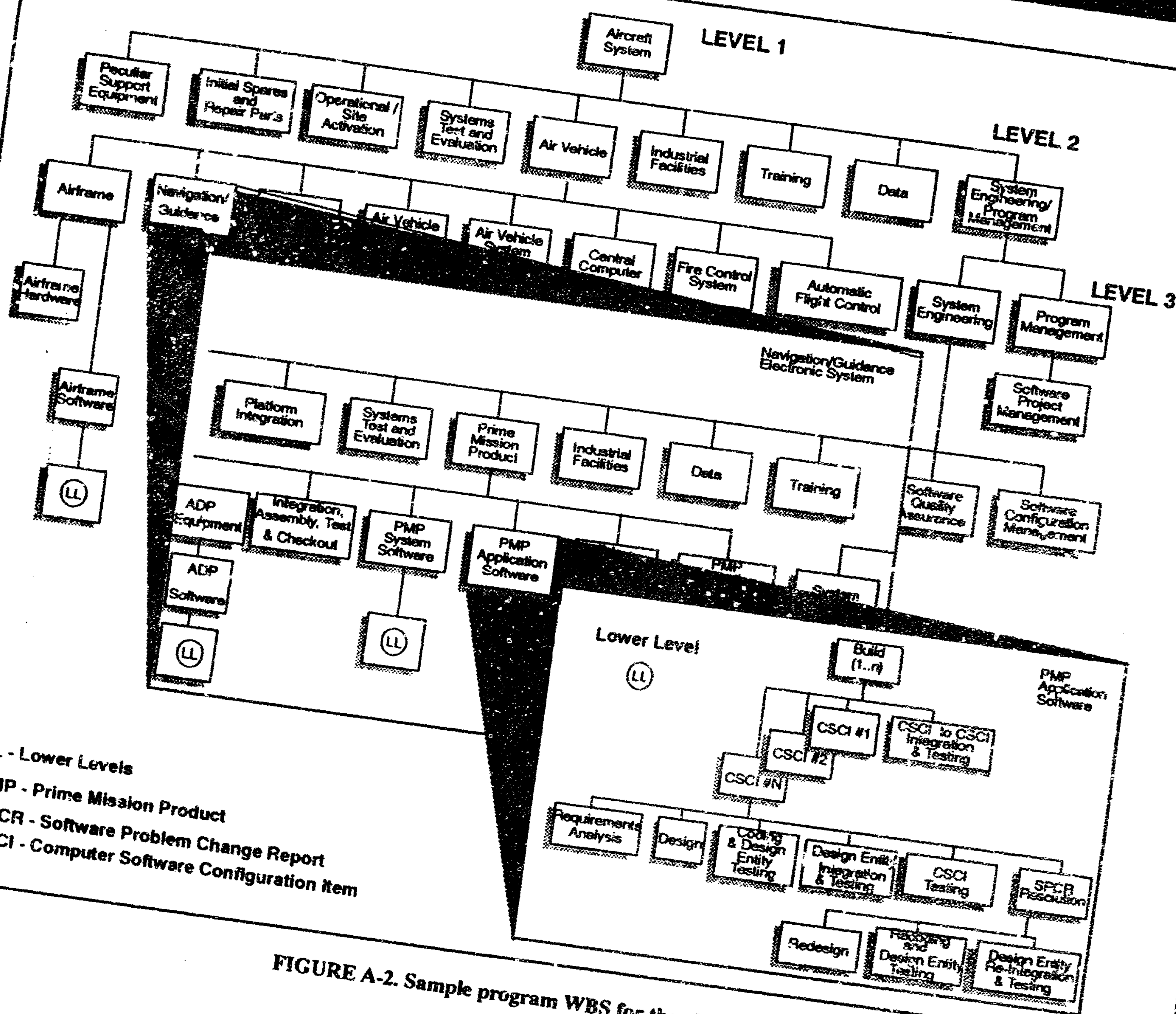
30.1.4.3 Stand alone software system. For this example, refer to Figure A-4, a supply system utilized by supply centers and depots is chosen. Since hardware was not part of the acquisition, the software system is the Primary Mission Product. This handbook will be used for acquisitions where the PMP is the software or software system and no hardware is included in the acquisition.

30.1.5 Correlate WBS to draft statement of work (SOW) and draft contract. The WBS development should be developed hand-in-hand with the draft SOW and the draft contract. A matrix should be developed which shows relationships between the WBS, the SOW, and the contract line item number (CLIN) structure. Another important part of the draft RFP which is related to the WBS is Instructions to Offeror. Within these instructions, the program office should explain which WBS items are mandatory, which items need to be extended, and what minimum level of break-out is required. The instructions should also include the requirement for the offeror to provide a matrix as described above.

An example of wording to include in the Instructions for Offeror is:

"Costs shall be broken out by WBS elements IAW the attached WBS. Offeror shall extend the WBS to the lowest level necessary to justify cost estimates. Offeror shall provide a matrix which correlates WBS elements with SOW tasks and the CLIN structure. These instructions apply only for proposal preparation and submission, they do not establish cost reporting requirements for post contract award efforts."

30.1.6 Tie the WBS to the cost/schedule reporting requirements. Work with the contracting officer, data manager and local cost analysis activity to ensure that the appropriate contract clauses and data items are tailored to reflect your information needs. Reference Contractor Cost Data Reporting pamphlet (AFLCP 800-15, AFSCP 800-15, AMC-P 715-8, NAVMAT P-5241)



LL - Lower Levels

PMP - Prime Mission Product

SPCR - Software Problem Change Report

CSCI - Computer Software Configuration item

FIGURE A-2. Sample program WBS for the aircraft system

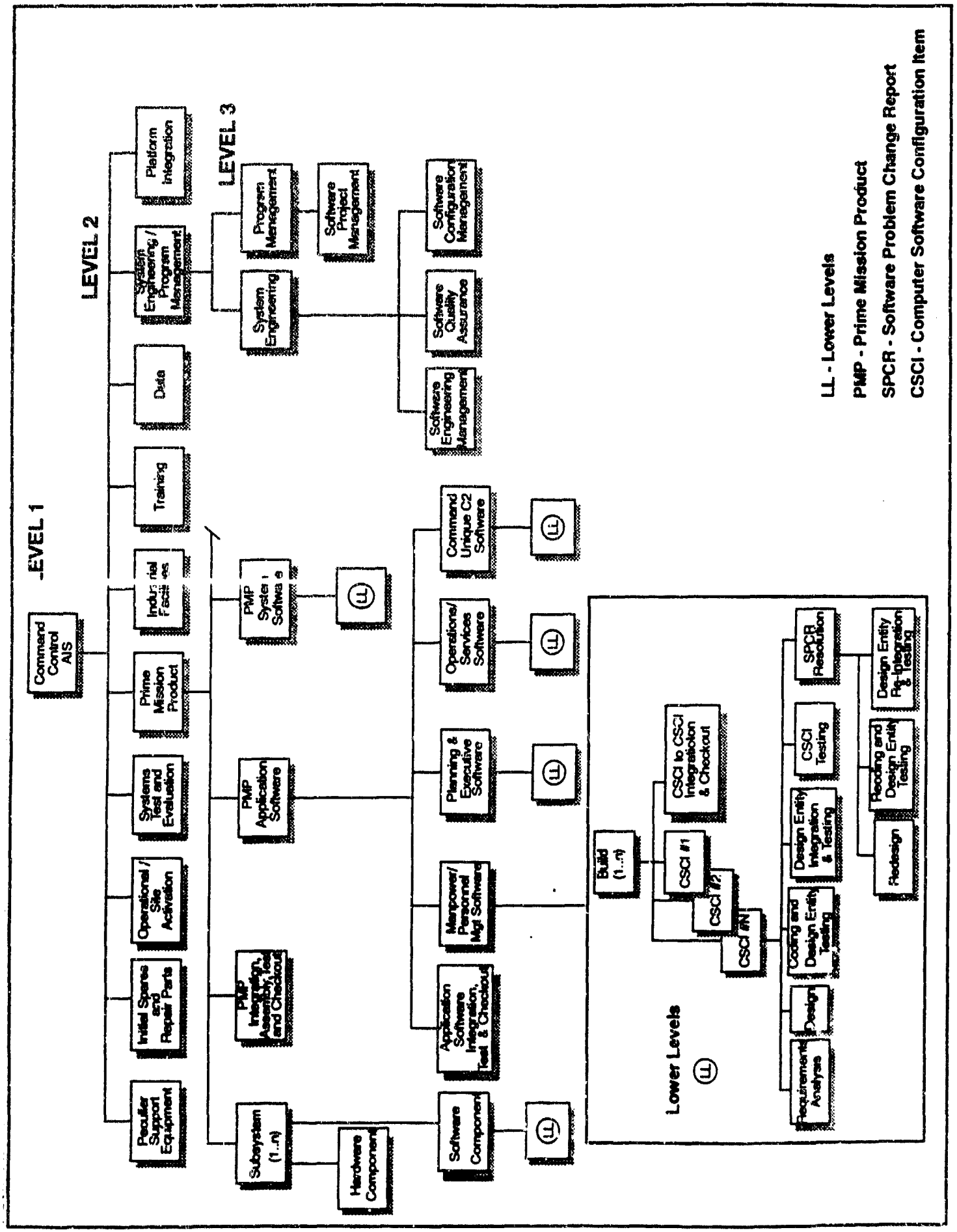


FIGURE A-3. Sample program WBS for Automated Information System

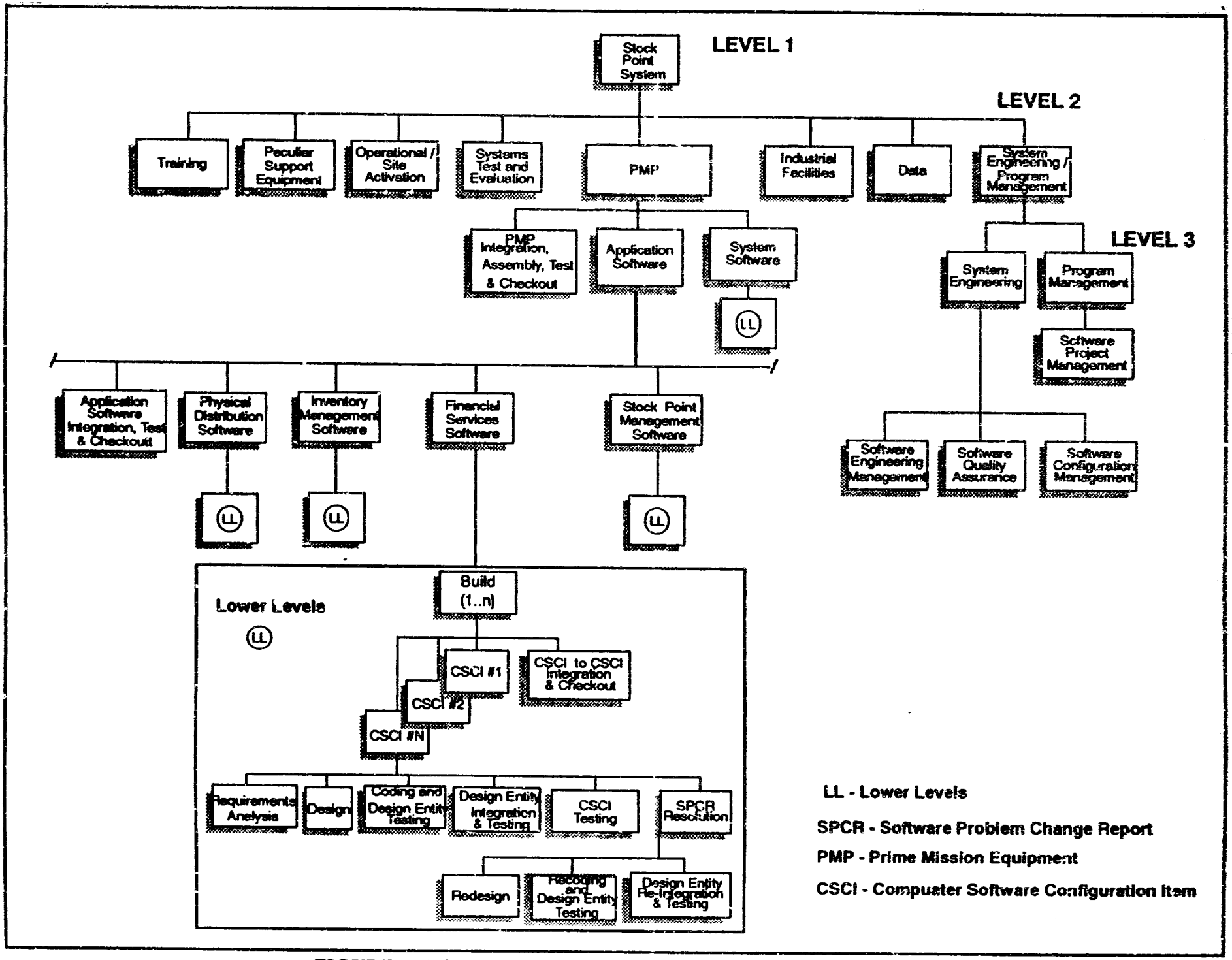


FIGURE A-4. Sample program WBS stand-alone Software System.

for specific guidance. Remember you should be requiring only the minimum amount of reporting necessary to achieve effective management control. Per MIL-STD-881 B, "... routine reporting is at level 3 of the contract WBS or higher, except for high-cost, high-risk, or other high-interest elements that are at lower levels." If a particular software item is not high-cost, high-risk, or high-interest, level 3 reporting should suffice. On the other hand, if the software item falls into one of the "high" categories, the program office must decide how low a level of reporting is necessary. Appendix B is a suggested approach which can provide the necessary parametric data and characteristics or attributes to perform cost estimates. These estimates can then be related to the WBS.SW to provide management insight into the SW development effort. In turn, actual information collected at the WBS.SW elements can be used to calibrate estimates for future efforts.

30.1.7 Send out a draft solicitation. After completing the above steps, and if time permits, send out a draft RFP for comments. This will provide the program office with feedback from industry regarding the WBS and reporting requirements.

30.2 Final solicitation preparation. Incorporate appropriate feedback from the draft solicitation, conduct appropriate coordination and reviews and distribute the final solicitation.

30.3 Proposal acceptance and source selection. In response to the program WBS, which includes the WBS.SW, contained in the Government solicitation, the contractor provides the contract WBS. The contract WBS identifies the framework for reporting program cost, schedule, and technical performance. The contractor will use this proposed WBS, including the WBS.SW, and expand it to lower levels. This provides Government software managers with a basis for uniform planning and reporting status. This also provides input which will be used for preparing the Cost Performance Report (CPR) and/or the Cost/Schedule Status Report (C/SSR).

30.4 Contract award. During contract negotiations, the WBS may be changed from the initial input. The final WBS is negotiated and incorporated into the contract. On completion of contract negotiation, the CCDR plan becomes the defining document for the exchange of cost data information. Refer to DODI. 5000.2 for specific guidance for preparing the CCDR plan.

30.5 Software element identifier. During reporting, to facilitate the roll-up of software cost data, an identifier for software related items is recommended. For example, the letter "SW" can be used as an extension to the reporting element number, as shown in Figure A-5. This is further explained in MIL-STD-881B, User's Guide Section II, paragraph 5.6.

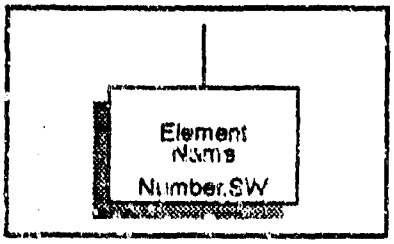


FIGURE A-5. Software element identifier

APPENDIX B

SOFTWARE CHARACTERISTICS DATA COLLECTION

10. GENERAL.

10.1 Scope. This appendix provides suggested guidance for collecting data on project characteristics related to project size, complexity, functions, schedule and level of effort. Software characteristics identified in this appendix are an appropriate set of focused data items that characterize the software being developed. These characteristics provide valuable data, which when collected and analyzed by the program manager (both government and industry), reinforces management indicators with credible software resource consumption estimates. The information contained herein is intended for guidance only.

10.2 Information. This appendix provides a suggested project questionnaire which may be used for collecting the project characteristics data. This information is important for providing project related data necessary to improve software acquisition planning, cost estimating, and cost estimating model calibration. The questionnaire, Software Parametric Data Collection Form, follows the appendix. This form is intended to be tailored. All information should NOT be required of all projects at all times. The software characteristics identified support input to most cost and schedule estimation tools.

20. APPLICABLE DOCUMENTS. This section is not applicable to this appendix.

30. Project/Computer Software Configuration Item (CSCI) Questionnaire Report. The Project/CSCI Questionnaire Report is contained on the following pages. The questions asked address both project specific data and CSCI(s) specific data.

30.1 Data maturity. As the project matures the software characteristic data collected/submitted are increasingly credible. Estimated responses should be indicated with an asterisk on the questionnaire.

40. Who needs the Data. The software characteristics data collected provides information required by cost/performance analysts, project software personnel, and project management personnel. The data can be used to continuously monitor and manage software development cost and schedule.

50. When to collect the data. The frequency at which the information is to be collected should be determined by the procuring activity based on, among others, the following criteria:

a. Risk (technical and management)

b. Schedule constraint
c. Cost constraint
d. Anticipated requirements volatility

50.1 Minimum reporting frequency. The data or sub-set may be collected during any point of the contract period of performance. However, it is suggested that at a minimum this data be collected:

a. To back up cost proposals for software costs.

b. At major software milestone reviews such as Preliminary Design Review (PDR), Critical Design Review (CDR)...etc., to be utilized by the procuring activity to update estimates of cost and schedule.

c. At the end of the contract as historical data. The software characteristics data will be used to improve future software acquisition planning, cost estimating, and cost model calibration.

50.2 Statement of Work (SOW) input. The data to be collected can be specified as a deliverable (DD Form 1423) in the contents of the SOW. It is suggested that DI-MISC-80048 Scientific & Technical Report Summary be used to provide this information

Technical Report Describing Contents of

MIL-HDBK-171 (Draft)

THIS PAGE IS INTENTIONALLY LEFT BLANK

PROJECT / CSCI QUESTIONNAIRE REPORT

GENERAL INFORMATION

Complete this form to the best of your ability for the project in question. If the question is not applicable, please mark it N/A. If you don't know the answer, leave it blank.

1. Firm or organization: _____
   Address: _____
   _____
   _____

2. Contract number: _____

3. Customer name: _____

4. Project overview description: (Should describe the CSCI being developed)
   To include: Functionality - expected impact of integrating COTS.

CSCI - Computer Software Configuration Item

A - PROJECT DESCRIPTION

1. Applications domain

[ ] Automation
[ ] Command & Control
[ ] Signal processing
[ ] Trainers
[ ] Interface systems
[ ] Data processing
[ ] Production control
[ ] Process control
[ ] Robotics/Mechanical
[ ] Manned Flight
[ ] Business
[ ] Communications
[ ] Test systems
[ ] Avionics
[ ] Graphic, image processing
[ ] Environments/Tools
[ ] Support software
[ ] Scientific
[ ] Unmanned flight
[ ] Other _____

2. Complexity

a. Rate The Difficulty Of Processing Logic

[ ] Simple processing logic, straight-forward I/O
[ ] Routine nesting, minimal interface with Operating systems, standard I/O
[ ] Difficult highly nested logic, real-time processing
[ ] Complex dynamic resource allocation, multiple exception handlers, recursion

b. Mathematical Complexity

[ ] Simple algorithms and simple calculations
[ ] Algorithms and calculations of average complexity
[ ] Many difficult algorithms and complex calculations
[ ] Majority of simple algorithms and calculations
[ ] Some difficult or complex calculations
[ ] Not Applicable

c. List the percent of total source code allocated to each of the following operational timing requirements:

Real-Time _____ %
Time-Constrained _____ %
On-line _____ %
Non-Time-Critical _____ %

d. Database Complexity

[ ] Simple data, few variables, low complexity
[ ] Multiple files, fields and data interactions
[ ] Highly complex
[ ] Simple, numerous variables
[ ] Complex file structure

3. Reliability

a. Most serious effect of a software failure

[ ] Inconvenience
[ ] Loss of human life
[ ] Major financial loss
[ ] Easily- recoverable loss
[ ] Moderate loss

b. Backup/Recovery considerations

[ ] Data protection beyond regular backup required
[ ] Alternative methods need to be developed in case of software failure
[ ] No special backup requirements

4. Is this the first system of its kind for your organization ?

[ ] Yes
[ ] No

5. Security Level

What are the security requirements on the development and on the target computing environments?

[ ] Class D
[ ] C1
[ ] C2
[ ] B1
[ ] None
[ ] B2
[ ] B3
[ ] Class A1
[ ] Other _____

6. Reused Code

a. Logical Complexity of Reused Code:

[ ] Simple algorithms and simple calculations
[ ] Algorithms and calculations of average complexity
[ ] Many difficult algorithms and complex calculations
[ ] Majority of simple algorithms and calculations
[ ] Some difficult or complex calculations
[ ] No reuse

b. Structural Complexity of Reused Code:

[ ] Nonprocedural (generated, query, spreadsheets, etc.)
[ ] Well structured (small modules and simple paths)
[ ] Poor structure with many complex paths and modules
[ ] Well structured with reusable modules
[ ] Fair structure but some complex paths and modules

c. Database Complexity

[ ] Simple data with few variables and little complexity
[ ] Multiple files, fields, and data interactions
[ ] Very complex data elements and data interactions
[ ] Several data elements but simple data relationships
[ ] Complex data elements and complex data interactions

d. Select the intended use of the majority of the software packaged for reuse:

[ ] None
[ ] Reuse across single product line
[ ] Reuse within single mission products
[ ] Reuse in any application

B - DEVELOPMENT METHODOLOGY

1. Milestones

Enter the expected and actual dates for each milestone below or N/A if the milestone does not apply to this project. If an expected date is an estimated date rather than a contract date, put an asterisk after that date. Provide this data for each incremental development.

Milestone	Expected Date	Actual Date
Project Start Date	_____	_____
System Requirements Review (SRR)	_____	_____
System Design Review (SDR)	_____	_____
Software Specification Review (SSR)	_____	_____
System Hardware Preliminary Design Review(PDR)	_____	_____
System Software Preliminary Design Review(PDR)	_____	_____
System Hardware Critical Design Revidw (CDR)	_____	_____
System Software Critical Design Review (CDR)	_____	_____
Test Readiness Review (TRR)	_____	_____
Functional Configuration Audit (FCA)	_____	_____
Physical Configuration Audit (PCA)	_____	_____
Formal Qualification Review (FQR)	_____	_____
Operational Test and Evaluation (OTE)	_____	_____
Project Completion Date	_____	_____

2. Schedule

a. Select the schedule and staffing constraints that best describe this development:

[ ] Normal average schedule, effort, and quality
[ ] Lowest cost with reduced staffing
[ ] Shortest schedule with high quality and reliability
[ ] Match staff size; with shortest schedule
[ ] Shortest development schedule, extra staffing
[ ] Highest quality and reliability
[ ] Match staff size; with normal development
[ ] Match staff size; with very high quality

3. Development paradigm employed (check all that apply):

[ ] Waterfall development
[ ] Phased builds
[ ] Rapid prototyping
[ ] Other _____
[ ] Incremental development
[ ] Spiral development
[ ] Pilot development

4. Software Reviews

a. Select all informal reviews held on the software during this development:

[ ] Design walkthroughs
[ ] Code walkthroughs
[ ] Other _____
[ ] Design inspections
[ ] Code inspections

5. System/Software Requirements

a. Select the option that corresponds to the level of definition and understanding of system requirements:

[ ] Very little definition and understanding of system requirements
[ ] Questionable definition and understanding of system requirements
[ ] Fairly complete definition and understanding of system requirements
[ ] Very complete definition and understanding of system requirements

c. Rate requirements volatility during development:

[ ] No changes
[ ] Small non-critical changes
[ ] Frequent non-critical changes
[ ] Occasional moderate changes
[ ] Frequent moderate changes
[ ] Many large changes

6. Design

a. Rate the maturity of the design concepts used:

[ ] Experimental
[ ] Evolutionary
[ ] State-of-the-art
[ ] Mature

b. Select any of the following design technologies, strategies, and tools used:

[ ] Applications Generator
[ ] Object-oriented methods
[ ] CASE tools
[ ] Structured analysis
[ ] Relational database
[ ] Query Language (SQL)
[ ] 4GL or 5GL
[ ] Reuse libraries
[ ] Exception handling
[ ] Front loaded scheduling
[ ] Management toolset
[ ] Cost and schedule models
[ ] Database management system
[ ] Small up-front design teams
[ ] Continuous integration via package specifications
[ ] Ada Program Design Language (PDL)
[ ] Other _____

7. Integration

Rate the expected level of difficulty of integrating and testing the CSCIs to the system level:

[ ] Very little integration, no complex interfaces
[ ] Average degree of complexity system integration and interface
[ ] Several system interfaces with some complex interfaces
[ ] Complex, time-intensive system integration process anticipated

C - SOFTWARE SIZE

1. Size Estimates (indicate if based on historical data or estimate, and use number if estimate)
   a. Number of software CSCI in this system: _____

   b. Select the basis for size estimates:
   [ ] Thousand of Source Lines of Code (KSLOCs)
   [ ] Function Points
   [ ] Other _____
   [ ] Carriage returns
   [ ] Semicolons
   [ ] Software Engineering Institute Method
   _____ (Specify)

   c. Enter the requested sizing information below, in thousands of units

CSCI	New	Size Reused	Modified	Language

2. Source Code Mix per CSCI

Source Code Type	% Code	% New Design	% New Code
Operating Systems			
Interactive Operations			
Real-Time Command & Control			
On-Line Communications			
Data Storage & Retrieval			
String Manipulation			
Mathematical Operations			
Other			

3. If the system was sized using function points, provide the following unadjusted function point information:
   a. Number of inputs (unique major data types that enter the system): _____
   b. Number of outputs (unique logical major report formats the system will generate): _____
   c. Number of inquiries (types of queries that result in informational searches and responses): _____
   d. Number of external interfaces: _____
   e. Number of inernal files (unique logical files/databases used by the application): _____

4. Database size (amount of code devoted to database definition accounts pluse databasse size in memory)
   Database size, as a percentage of total program size: _____ % (source lines of code)

D - PROJECT STAFFING

1. Staff Size
   a. Minimum staff size: _____

   b. Maximum staff size: _____

2. Staff Skill/Experience
   a. Average Analysts' experience with similar applications: _____ years _____ months

   b. Average Programmers' experience with this host machine: _____ years _____ months

   c. Average Analysts' experience with chosen development Methodology: _____ years _____ months
      [ ] Completely new development methodology
      [ ] Minor experience with development methodology
      [ ] Extensive experience with development methodology

   d. Average Analysts' experience with the Development language: _____ years _____ months
      [ ] Completely new development language
      [ ] Educational or training in language
      [ ] Some experience with development language
      [ ] Extensive experience with development language

   e. Programmers' Average Ada Environment Experience: _____ years _____ months

   f. Development organization's experience developing this type of application
      [ ] This application is a new project not in our current line of business
      [ ] This application is a normal development project that is part of our current line of business
      [ ] This application is a familiar type of project having already been developed by the company before or similar to other projects we have developed
      [ ] Many applications of this type have been developed by the company (greater than 7)

E - COMPUTER SYSTEM

1. Development Environment

   a. Rate the virtual machine volatility of the development system based on frequency of major/minor changes:

   [ ] 12 months (major)/1 month (minor)
   [ ] 6 months/2 weeks
   [ ] 2 months/1 week
   [ ] 2 weeks/2 days

   b. Select the following option that best assesses the embedded features of the development systems:

   [ ] Hardware is to be developed, but its completion will occur long before the software is to be ready
   [ ] Hardware is to be developed on the contract, it is to be developed concurrently with the software and the hardware can/does have major impacts on the software
   [ ] Hardware is to developed on the contract, it is to be developed concurrently with the software but the hardware has little impact on the software
   [ ] No new hardware is to be developed under the effort; there will be no impact on the software development

2. Target Computer Configuration (complete this section only if the development system differs from the target

   a. Rate the virtual machine volatility of the target system, based on number of major/minor changes:

   [ ] 12 months (major)/1 month (minor)
   [ ] 6 months/2 weeks
   [ ] 2 months/1 week
   [ ] 2 weeks/2 days

   b. Identify the system architecture:

   [ ] Centralized (single processor)
   [ ] Tightly-coupled (multiple processor)
   [ ] Loosely-coupled (multiple processor)
   [ ] Functional processors communication via a bus
   [ ] Distributed (centralized database)
   [ ] Distributed (distributed database)

3. Performance Requirements

   a. Main Storage Constraint. Main storage refers to direct random access storage such as core, integrated-circuit, or plated-wire storage; it excludes such devices as drums, disks, tapes,or bubble storage. Indicate the percentage that best reflects the main storage expected to be used by any sub-systems consuming the main storage resources. _____ %

   b. Overall Target Hardware Constraints. Overall hardware refers to processor memory, speed, and throughput

   [ ] Close to 100% utilization
   [ ] Difficult hardware capacity limitations (75% to 90%)
   [ ] Average hardware capacity limitations (60% to 75%)
   [ ] No hardware capacity limitations (less than 60%)

F - DEVELOPMENT ENVIRONMENT

1. Project Organization

a. Check all of the company organizations included in effort estimations/actuals

[ ] Systems engineering
[ ] Marketing
[ ] Program management
[ ] Configuration management
[ ] Data management
[ ] Database administration
[ ] User department
[ ] Software development
[ ] Software test
[ ] Quality assurance
[ ] Independent V&V
[ ] Other _____

b. Organizational Interface Complexity

[ ] Single customer, single interface
[ ] Multiple internal and external interfaces
[ ] Multiple internal interface single external interface
[ ] Single customer co-located with developer
[ ] Multiple interfaces geographically distributed

c. Multiple site development

[ ] Single site and single organization
[ ] Multiple sites, same general location
[ ] Single site and multiple organizations
[ ] Multiple sites, located 50 miles or more apart

2. Computer Resources

Select the percentage of time that the development computer is available for use on this project:

[ ] 10%
[ ] 70%
[ ] 40%
[ ] 100% (fully dedicated)